Large-Scale 3D Environmental Modelling and Visualisation for Flood Hazard Warning

Chen WANG

A thesis submitted in partial
fulfilment of the requirements for the
degree of Doctor of Philosophy

Department of Creative Technology

School of Computing, Informatics
and Media

University of Bradford

2009

Abstract

3D environment reconstruction has received great interest in recent years in areas such as city planning, virtual tourism and flood hazard warning. With the rapid development of computer technologies, it has become possible and necessary to develop new methodologies and techniques for real time simulation for virtual environments applications. This thesis proposes a novel dynamic simulation scheme for flood hazard warning. The work consists of three main parts: digital terrain modelling; 3D environmental reconstruction and system development; flood simulation models. The digital terrain model is constructed using real world measurement data of GIS, in terms of digital elevation data and satellite image data. An NTSP algorithm is proposed for very large data assessing, terrain modelling and visualisation. A pyramidal data arrangement structure is used for dealing with the requirements of terrain details with different resolutions. The 3D environmental reconstruction system is made up of environmental image segmentation for object identification, a new shape match method and an intelligent reconstruction system. The active contours-based multi-resolution vector-valued framework and the multi-seed region growing method are both used for extracting necessary objects from images. The shape match method is used with a template in the spatial domain for a 3D detailed small scale urban environment reconstruction. The intelligent reconstruction system is designed to recreate the whole model based on specific features of objects for large scale environment reconstruction. This study then proposes a new flood simulation scheme which is an important application of the 3D environmental reconstruction system. Two new flooding models have been developed. The first one is flood spreading model which is useful for large scale flood simulation. It consists of flooding image spatial segmentation, a water level calculation process, a standard gradient descent method for energy minimization, a flood region search and a merge process. The finite volume hydrodynamic model is built from shallow water equations which is useful for urban area flood simulation. The proposed 3D urban environment reconstruction system was tested on our simulation platform. The experiment results indicate that this method is capable of dealing with complicated and high resolution region reconstruction which is useful for many applications. When testing the 3D flood simulation system, the simulation results are very close to the real flood situation, and this method has faster speed and greater accuracy of simulating the inundation area in comparison to the conventional flood simulation models.

Acknowledgments

I would like to express my sincere gratitude to my supervisor Dr. T. R. Wan. Thank you for your heuristic ideas, constructive suggestions, patient instructions and continuous support during my PhD research. You have successfully kept my research progress and led my work to a right direction. Without your help, my research would never go smoothly, or even be possible.

I am very grateful to my second supervisor Dr. Ian. J. Palmer, whose ideas and discussion have benefited me a lot.

I would also like to deliver my grateful thanks to my aunt Sandy Johnson. Thank you for your selfless and endless love.

More thanks to my colleagues in Visual Computing Group, for all your assistance in my life and study, which made me feel like in a big warm family.

Finally, I would like to give my deepest thankfulness to my dear parents, who are far away in China, but always care about me, encourage me and try their best to support me. This thesis is dedicated to them.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

NTSP	**Nona-Tree Space Partitions**
1D/2D/3D	**One-Dimensional / Two-Dimensional / Three-Dimensional**
AC	**Active Contour**
GIS	**Geographic Information System**
DTM	**Digital Terrain Model**
LOD	**Level of Detail**
DEM	**Digital Elevation Model**
SRTM	**Shuttle Radar Topography Mission**
ROAM	**Real-time Optimally Adapting Meshes**
USGS	**U.S. Geological Survey**
VDPM	**View Dependant Progressive Meshes**
DAG	**Directed Acyclic Graph**
CAD	**Computer-aided Design**
DSM	**Digital Surface Model**
TIN	**Triangle Irregular Net**
UTM	**Universal Transverse Mercator**
LL	**Latitude Longitude**
LIDAR	**Light Detection and Ranging**
SAC	**Spectral Angle Classifiers**
NVDI	**Normalized Difference Vegetation Index**
EVI	**Enhanced Vegetation Index**
MEL	**Maya Embedded Language**
FVM	**Finite Volume Method**
FEM	**Finite Element Method**
FDM	**Finite Difference Method**

Chapter 1 Introduction

Flood hazard warning has received increasing attention in recent years. The key issues regarding flood hazard warning include implementing a user-friendly, interactive decision support system, identifying the affected areas and using a dynamic flood spreading modelling. Many flood warning systems were developed for 2D environments and limited on specific flood hazard. In order to overcome these disadvantages, new methodologies and techniques for 3D real time flood simulation is necessary. Our methodology was developed from this point of view for modelling and reconstructing a 3D real world and simulating general flood spreading progress in large-scale or dense regions to identify the potential inundation area effectively.

This research work achieved following main objectives:

1) The new dynamic terrain modelling method which is capable of implementing out-of-core rendering and dealing with massive Geographic Information System (GIS) terrain data.
2) A multi-resolution vector-valued framework for image segmentation based on active contours is defined which has the advantage of dealing with complicated colour images rapidly and precisely.
3) A new strategy to design the 3D environment reconstruction system which is suitable for both large-scale and high resolution environment reconstruction based on specific segmented areas and shape match method.

4) Two new flooding models for 3D flood simulation have been developed which can handle dynamic flood behaviour and predict inundation areas in real time.
5) A new 3D flood simulation system prototype for flood hazard warning has been developed in order to let governments make timely and accurate decisions for the protection of lives and property.

1.1 Background

Large-Scale flooding is a serious natural disaster that frequently occurs in many parts of the world, particularly in America, Asia and Europe. In 2005, a horrible flood caused by the hurricane Katrina nearly destroyed the whole of New Orleans in America. More than twenty thousand people suffered the consequences of the flood and about forty thousand houses were submerged. The cost of the flood damage to New Orleans was approximately $22.6 billion. In China in 2002, a flood reached more than 20 provinces, damages have been estimated to $8.2 billion, and 110 million people have been affected with over 1500 deaths. In the UK, in August 2002, York and Pickering were hit by a flood that was a result of very heavy rainfall, in which some areas the rain exceeded the usual monthly amount in less than 24 hours. Many governments are making huge efforts and using different material resources as well as spending huge amount of money to construct flood defence systems and strategies for natural disaster prevention.

Currently, flood modelling and simulation play a very important role and make a real contribution to the work against flood hazards. Some efficient flood modelling and post-flood damage estimation systems have been designed for solving or mitigating the loss [Haile and Rientjes 2005] [Gianinetto et al 2006].

The system developed lacked the support of GIS data, and it would be difficult to apply it to flood inundation mapping. At the same time, it was developed for 2D environment and was not integrated with a large-scale 3D environment.

Due to the intuitive result, large-scale 3D environmental modelling is becoming more and more popular. Over the past few years, it has become increasingly common to use image-based reconstruction techniques on realistic terrain models [Sequeira et al 2001] [Debevec 2003] [Hakim 2002]. A fast process for large data set is required in order to present terrain in real-time and reconstruction techniques for modelling the real world remain the bottleneck.

In order to overcome these problems, a new flood hazard warning system has been proposed which includes digital terrain modelling; 3D environmental reconstruction and system development, flooding models. Digital terrain modelling is used to perform out-of-core rendering and to deal with massive GIS terrain data with an NTSP algorithm and to process and present the detailed terrain information with high resolution data set through an in-core pyramidal data structure. For environmental quality between large areas and the detailed level, a 3D environmental reconstruction system is proposed by using 3D virtual reality models, image segmentation for object identification, a shape matching method and an intelligent reconstruction system. Regarding the flood simulation, two new flooding models have been developed: a flood spreading model and a finite volume hydrodynamic model. The first model is useful for large scale flood simulation. The second model is built from shallow water equations which is useful for urban area flood simulation. A combination of the two flooding models can predict inundation area more accurately.

1.2 Motivation

Some terrain modelling work is based on artificial data which can not represent a real 3D environment. In August 2004, Shuttle Radar Topography Mission (SRTM) data became available via the ESDI website which is free elevation data for education. By means of this data, a 3D real world can be constructed. This is a strong stimulation for doing this research.

On the other hand, natural disaster prevention has been a worldwide topic and many governments spend a lot of financial and material resources in constructing disaster defence systems, especially for flooding prediction. There is a need to achieve an efficient, practical prediction system and to decrease the expenditure on such systems. Traditional methods for forecasting and detecting floods are only based on 2D satellite imagery. The build up of a heavy cloud during high rainfall events may make it difficult or impossible to use conventional applications of aerial photography or Landsat data for observation of inundation patterns [Sippel et al 1994] [Yamagata and Yasuoka 1993]. Ground measurement of flooding in forested wetlands is also severely limited by the inaccessibility typical of these areas, where mobility is often hampered by flooding and boggy conditions. Remote detection of flooding is hampered by forest canopies that render the land/water boundary invisible to infrared and visible wavelength sensors and by frequent cloud cover during periods of rainfall [Sippel et al 1994].

By making use of a real 3D environment which we incorporate with the flood region spreading model and the finite volume hydrodynamic model, we can compute the flood speed and direction in order to simulate a real flood disaster.

1.3 Research Problems

Our project includes four major parts: digital terrain modelling, image segmentation based on active contours, 3D environmental reconstruction and 3D real time flood simulation.

There are two aspects to construct the real world: generating 3D terrain from digital data and implementing map projections. The problem of the first part involves mesh presentation, level of detail (LOD) method selection and large dataset structure arrangement. For implementing map projections, matching of each pixel of Landsat maps with the generated meshes accurately is another difficulty. At present time, almost all GIS systems are based on a Gaussian coordinate system, which is a Gauss plane (a linear space), however since the earthly space is nonlinear, the spatial reference must be addressed by the projection.

Modelling the digital terrain is the first step in trying to perform real time rendering. As we know we cannot put all the image data in the computer memory especially for large data sets. The solution for this problem is to perform dynamic rendering dependent on the viewer's position.

Thirdly, when an observer moves close to the earth surface region for the details, it is necessary to use a detailed or high-resolution data set for the environment reconstruction.

Fourthly, in order to reconstruct an urban environment, precise object segmentation for objects such as buildings, roads and vegetation is very important. With the aim of flood simulation, river recognition is a strong requirement.

The final issue is how to construct the efficient model based on the real world environment to simulate the actual situation of flood disaster in real time.

1.4 Aim and Objectives

The main focus of this project is to generate and reconstruct large scale urban environments from satellite images and real elevation data and apply this to construct an efficient 3D dynamic simulation scheme for flood hazard prediction.

Most of the existing research on urban reconstruction has focused on creating aesthetically pleasing models for virtual tours and other entertainment purposes. Our purpose is to create models that can be used in applications such as emergency simulation, flood simulation, disaster preparedness, urban planning and change monitoring and environmental resource management. We propose to develop algorithms for automatically creating these models. The models will then be used to reconstruct the whole 3D urban environment for both large scale and highly detailed small scale. Prior to this stage, automatic detection of buildings, vegetation and roads from aerial imagery needs to be preprocessed.

Large numerical calculations are required to make a prediction of the possible damage a flood would cause. The results of these simulations can be used to prevent further flood catastrophes. The realistic 3D visualization of the resulting data is very important. The more realistic a visualization is the more useful the precaution that can be taken by the local authority and the citizens. The goal of flood simulation is to build a platform which allows a realistic looking 3D dynamic visualization of flood disaster in real-time for hazard prevention and risk assessment.

1.5 Main Contributions

The contributions of this research work consist of:

A flexible dynamic terrain modelling method: this is made up of a Nona Tree Space Partitions (NTSP) algorithm using satellite images, elevation and GIS data and a pyramidal data structure. NTSP is proposed for very large data assessing, terrain modelling and visualisation. The pyramidal data structure is used for dealing with the requirement of terrain detail with different resolutions.

An efficient multi-resolution vector-valued framework for image segmentation based on active contours is used. It includes a vector-valued scanning algorithm, a geometric active contours model, low resolution segmentation with elevation data and high resolution optimization. Our approach has the advantages of dealing with complicated satellite images, automatically initializing a number of snakes based on colour and texture features, accurately and rapidly identifying the target objects with elevation data.

A novel 3D environmental reconstruction approach is used: 3D large scale environmental reconstruction method is composed of our 3D digital real environment generation algorithm**,** environmental image segmentation using a geometric active contours model and a multi-seed region growing method and an intelligent reconstruction system. The 3D detailed urban environment reconstruction approach includes basic types of primitive, detailed building detection and an intelligent shape match method. Our method is capable of dealing with complicated and high resolution region reconstruction which is useful for many applications.

Two new flood simulation models are used:

(1) The flood spreading model that is based on flooding image spatial segmentation, a water level calculation process, a standard gradient descent method for energy minimization and a flood region search and merge process.

(2) The finite volume hydrodynamic model that is built from 2D shallow water equations which is useful for urban area flood simulation.

Simulations have been carried out and show that the two models are suitable for realistically modelling and visualising flood hazard events for early warning and prediction.

The architecture of the proposed large scale environmental generation and reconstruction system framework and its application for flood simulation are shown in Figure1.1.

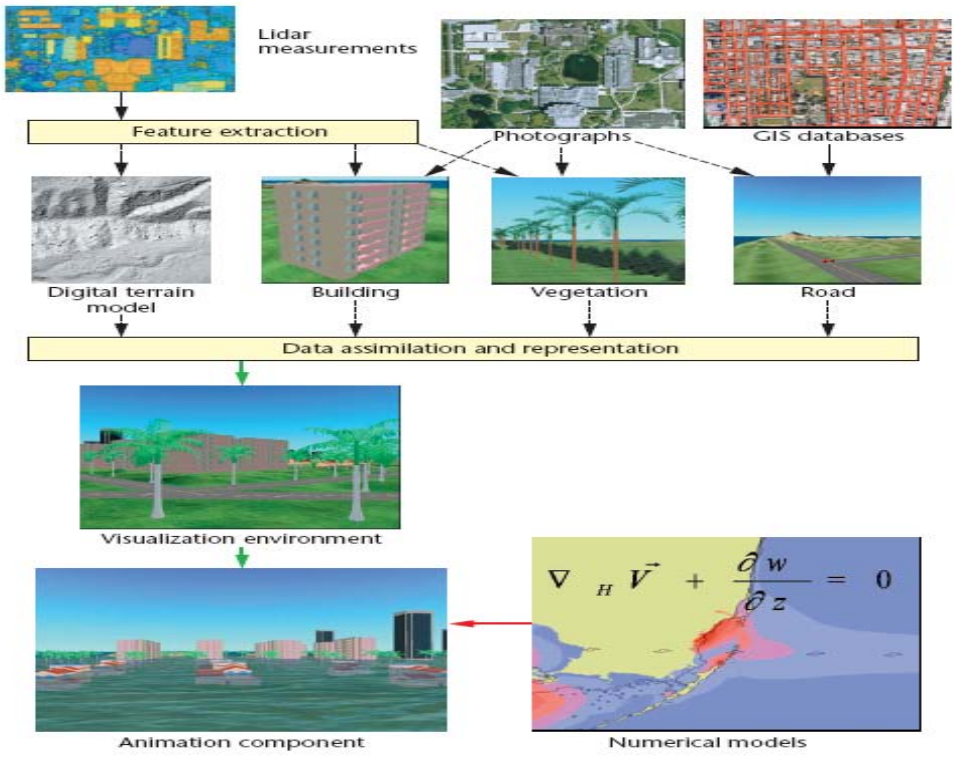


Figure 1.1 Framework for 3D visualization and flood simulation

1.6 Thesis Structure

The rest of this thesis is organised as follows:

Chapter 2 reviews previous and current methodologies and applications on: (i) Digital terrain model generation; (ii) Environmental image segmentation; (iii) 3D environmental reconstruction and (iv) 3D flood simulation. All of these four research areas present a challenge to the researchers.

Chapter 3 presents a new dynamic terrain modelling with an NTSP algorithm and a pyramidal data structure.

Chapter 4 introduces several image processing techniques and reviews some typical image segmentation methods. After that, a novel multi resolution vector valued framework for image segmentation based on active contours is described. This framework consists of vector valued scanning algorithm, geometric active contours model, low resolution segmentation with elevation data and high resolution optimization.

Chapter 5 explains our method for 3D large-scale environmental reconstruction and 3D high resolution environmental reconstruction.

Chapter 6 discusses our simulation system architecture that includes a 3D urban environment construction system and a 3D flood simulation system. Our simulation platform offers convenient methods for constructing the real urban environments, specifying the flood hazard prevention tasks and monitoring the inundation areas

Chapter 7 describes and discusses two new flood simulation models for flood hazard warning and prevention which are important applications to link with our 3D environmental reconstruction.

Chapter 8 describes specific flood simulation cases for testing our flood warning system. They are not only implementing a flood spreading process within 3D environment for predicting the potential inundation area, but also providing useful information to make the right decision when a real flood occurs.

Chapter 9 summarises the thesis and proposes some ideas for future work.

The appendix provides 3D real world modelling code, image segmentation based on active contours code, shape recognition in satellite image code, 3D environment reconstruction and flood simulation code and list of publications.

All sources of reference are provided in the end of the thesis.

Chapter 2 Literature Review

2.1 Introduction

The task of flood hazard warning is to model and reconstruct the 3D real world and simulate flood spreading progress in order to identify the potential inundation area. This chapter discusses the previous and current methodologies and applications in these areas which present a challenge to the researchers.

This chapter is organized as follows:

- Section 1 introduces the classic methods of digital terrain modelling.
- Section 2 reviews image segmentation and recognition algorithms.
- Section 3 describes the approaches of 3D environment reconstruction.
- Section 4 explains some typical methods of 2D and 3D flood simulation.
- Section 5 draws the conclusion.

2.2 Methods of Digital Terrain Model Generation

The digital terrain model (DTM) is a digital representation of ground surface topography or terrain.

Digital terrain models may be prepared in a number of ways, but they are obtained frequently by remote sensing rather than direct survey. One powerful technique for generating digital terrain models is interferometric synthetic aperture radar; two passes of a radar satellite (such as RADARSAT-1) suffice to generate a digital elevation map tens of kilometres along with a resolution of around ten meters. This also obtains an image of the surface cover.

Another powerful technique for generating a DTM is using the digital image correlation method. It requires two optical images acquired from different angles

taken from the same pass of an airplane or an Earth Observation satellite (such as the HRS instrument of SPOT5).

Older methods of generating DTM often involve interpolating digital contour maps that may have been produced by direct survey of the land surface; this method is still used in mountain areas, where interferometry is not always satisfactory. A DTM implies that the elevation is available continuously at each location in the study area.

The quality of a DTM is a measure of how accurate the elevation is at each pixel (absolute accuracy) and how accurately is the morphology presented (relative accuracy). Several factors play an important role for quality of DTM-derived products:

- Terrain roughness;
- Sampling density (elevation data collection method);
- Grid resolution or pixel size;
- Interpolation algorithm;
- Vertical resolution;
- Terrain analysis algorithm;

There are many virtual environments constructed using artificial terrain. The important issue is how to render a 3D real model whilst solving practical difficulties. SRTM provides real data image for one point in 90m×90m area. We can use real data instead of a greyscale map which is a critical component of large-scale geometrical environment generation. In the following section we will describe related work in large-scale environment visualization. We will focus particularly on real time 3D real world rendering algorithms.

2.2.1 Real-time Continuous Level of Detail Rendering of Height Fields

Lindstrom's algorithm [Lindstrom et al 1996] is a block based one that uses a bottom-up strategy. It first assesses the level of the block according to a screen projection error metric, and then it inserts those vertices which have a delta value larger than the threshold. To eliminate cracks between adjacent nodes with different levels, Lindstrom uses a binary vertex tree to maintain the dependence between vertices. For cracks between blocks, Lindstrom suggests adjacent blocks share vertices on their boundaries. The algorithm can exploit the coherence between frames to reduce the number of vertices processed per frame. It generates resultant meshes based on quadtree subdivision with the vertices chosen in the previous step. The algorithm uses 6 to 28 bytes per vertex; which is not high compared to Hoppe's progressive meshes. The screen space error metric as proposed by Lindstrom was widely used or adapted by various terrain rendering algorithms, including ROAM [Duchaineau et al 1997], Hoppe [Hoppe 1997] etc. It proves to be an effective way to evaluate the importance of vertices in the original sampling grid. The terrain data Lindstrom used were two large high-resolution one (2m×2m for 64 km2 and 1m×1m for 14 km2). The result shows that the algorithm can work well with large-scale high-resolution terrain data.

2.2.2 Real-time Optimally Adapting Meshes

One year later, in 1997, Duchaineauy et al [Duchaineau et al 1997] applied ROAM in his terrain with Real-time Optimally Adapting Meshes. ROAM, which has been widely used in games, was based on Lindstrom's algorithm but it is much faster. It also performs triangle splitting and merging based on triangle diamonds (Figure 2.1). It uses an explicit binary triangle tree structure, which is

similar to Lindstrom's binary vertex tree. However, whilst Lindstrom's approach emphasized a bottom-up algorithm, ROAM uses a splitting diamond queue and a merging diamond queue to progressively make updates to the triangle mesh. For both queues, triangle diamonds are ordered by their priorities which can be calculated using delta values as well as other error metrics. The algorithm is error bound; it guarantees the priority to be monotonous while doing a top-down subdivision. Moreover, ROAM is a strict error control; it can have an explicit control over triangle numbers and LOS (Line of Site). The algorithm exploits more of the coherence between frames and can maintain rather more stable frame rates. The terrain data which is being used is a USGS 1-degree (90m spacing) Degital Elevation Model (DEM) data. It is relatively crude compared to the data Lindstrom used. Although the ROAM was widely used in games, Jonathan [Jonathan 2000] concluded that for dense data it is "not too great" after several months of work.

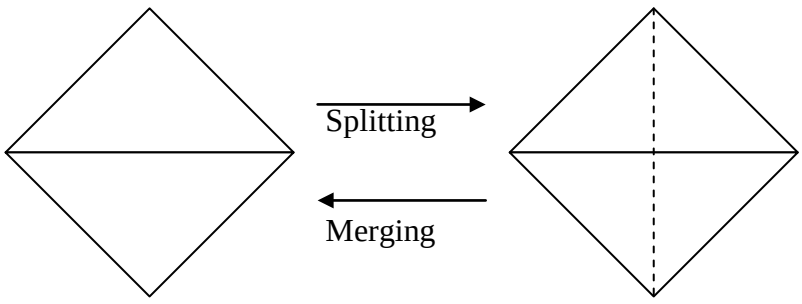


Figure 2.1 splitting and merging process

Although Röttger gave his paper [Röttger 1998] almost the same title as Lindstrom's, Röttger managed to use a different algorithm from Lindstrom's —— in fact in his paper, he presented an important method of crack eliminating which has since been widely used. Röttger's algorithm is regular grid based too, however he uses an error metric different from Lindstrom's, which uses a hybrid of

distances from the viewpoint and local delta values. To eliminate cracks he uses not only cracks but also the dependence between adjacent quadtree nodes. He developed a way of error calculation that guarantees that the maximum difference in levels between adjacent nodes will never exceed 1, and thus reduced the work to do for crack elimination. Röttger's algorithm also supports geomorphing that can reduce or even eliminate the vertex popping in consequently rendered scenes that Lindstrom's approach suffers.

2.2.3 Terrain Rendering Techniques

In 1998, Hoppe [Hoppe 1998] experienced real-time fly-over by applying his View Dependant Progressive Meshes (VDPM) on terrain. Hoppe uses a 4097×2049 Grand Canyon DEM data (60m spacing) with an orange looking, 4096×2048 texture. Hoppe divided the terrain into several blocks since the whole terrain is too large to fit in a single Progressive Mesh structure. For each of the blocks, Hoppe generates TINs with VDPM according to viewpoints and an error threshold. Blocks share boundary vertices to join them seamlessly. Hoppe's VDPM can support geomorphing as well. He uses screen space error as the error metric but simplified Lindstrom's formulas to the following:

$$\Delta h = k(v - e) \cdot \vec{e} \quad k = \left(\frac{\tau}{\lambda}\right) * \left(\frac{1}{d}\right) \tag{2.1}$$

where e is the viewpoint, v is the screen space error threshold, d is the distance between viewpoint and screen, λ is the number of pixels per world coordinate unit. Though Hoppe's algorithm performs well with the data, it is a TIN based one that demands too much storage and suffers from too many constraints——it lacks generality. Though it can produce a mesh with far less triangles than a regular grid based one, optimization takes too much time.

Real-time terrain rendering is mostly used in games and it is not surprising that many game companies are more adept in the field of Computer Graphics than research institutes. In SIGGRAPH 2000 course 39, Jonathan from Bolt-Action Company made a brief slide show [Jonathan 2000] on terrain rendering at high levels of detail. He discussed in detail the High Levels of Detail Terrain Rendering in [Lindstrom and Pascucci 2002], where he pointed out that —— with their experience —— for densely sampled data, ROAM doesn't perform very well. Jonathan ascribes the decline in performance to the priority re-computation of the triangles in merging and splitting queues (though only a small portion of triangles need to be re-computed, the absolute number increases when there are a large number of triangles in the queue.). Jonathan also points out the traditional screen space error metric has an inherent shortcoming —— it discards 2 of the 3 dimensions of a vertex when performing a 3D to 1D projection. To avoid this shortcoming, Jonathan uses an isosurface LOD test for every vertex instead of evaluating every vertex's error at each viewpoint. The simplest form of an isosurface is isosphere, and once the viewpoint enters the isosphere the vertex will need to be inserted, otherwise it need not be. Different isospheres can form hierarchies and it speeds up the level of detail control.

2.2.4 Terrain Simplification Method

In 2002, Lindstrom Peter and Pascucci Valerio [Lindstrom and Pascucci 2002] extended their previous work to allow view-dependent out-of core simplification and refinement visualization.

In this paper, they provide two important components of terrain framework: view-dependent refinement of the terrain mesh and a simple terrain data organization from external storage to main memory. They utilise a special type of

subdivision based on longest edge bisection. The mesh produced by this subdivision can be refined locally without having to maintain the entire mesh at the same resolution. Whether to split an edge is generally based on whether the vertex's diamond approximates the corresponding part of the full-resolution mesh well enough. For view-dependant refinement, they adopt a nested Directed Acyclic Graph (DAG) [Thomas et al 1990] sphere. Each sphere is centred on the position pi of a mesh vertex *i*, and represents the isocontour of *i*'s projected screen space error $\rho i = \rho$ (εi, *pi*, *e*), where εi is an object space error term for *i* and *e* is the viewpoint. The formula they use is:

$$\rho\ (\varepsilon_i, p_i, e) \geq \rho\ (\varepsilon_j, p_j, e) \quad \forall_j \in C_i \tag{2.2}$$

This means that *i* is active when the viewpoint is inside the sphere. The authors in [Lindstrom and Pascucci 2002] use three kinds of error metrics: object space error metrics, isotropic error projection and anisotropic to compute actual screen space errors, then to construct their top-down, recursive refinement and on-the-fly triangle strip model.

In order to lay out the terrain data on disk to achieve out-of-core performance, they apply two types of data structure to the straightforward implementation: interleaved quadtrees and embedded quadtrees. Through this method, they can minimize paging events and compute the index of the data element in the given refinement efficiently.

One weakness of their approach is that the height field representation created by edge bisection is necessarily restricted to be of dimensions $(2n + 1)^2$. Another constraint in their current system is that the input data for a single grid sampled should be of uniform resolution.

2.3 Methods of Image Segmentation and Recognition

Segmentation refers to the process of partitioning a digital image into multiple regions. The goal of segmentation is to simplify or change the representation of an image into something that is more meaningful and easier to analyze.

2.3.1 Edge Based Segmentation

Edge based segmentation represents a large group of methods based on information about edges in an image; it is one of the earliest segmentation approaches and still remains very important. Edge-based segmentation relies on edges found in an image by edge detecting operators. A variety of edge detecting operators have been developed, after an image is processed by edge detecting operators, supplementary processing steps must follow to combine the edges into edge chains, and the final result is to group local edges into an image where only edge chains with a correspondence to existing objects or image parts are presented.

The simplest method is the edge image threshold. Hanson and Riseman proposed a classical edge context evaluation method [Hanson and Riseman 1978]. This method uses crack edges (edges located between pixels), which produce some favourable properties. Often there is a situation where a region border is unknown but the region has been defined in the image. A method has been suggested to find the inner border and the outer border [Milan et al 1998], then a boundary whose properties are better than the inner and outer borders called extended borders can be found in the adjacent regions, thus the borders can be uniquely detected. Based on this idea different methods to extract extended borders were proposed [Liow 1991]. An example is presented for a typical edge

detection result. Figure 2.2 presents the original image and Figure 2.3 presents the outcome of finding the edges of an image using the Canny method.

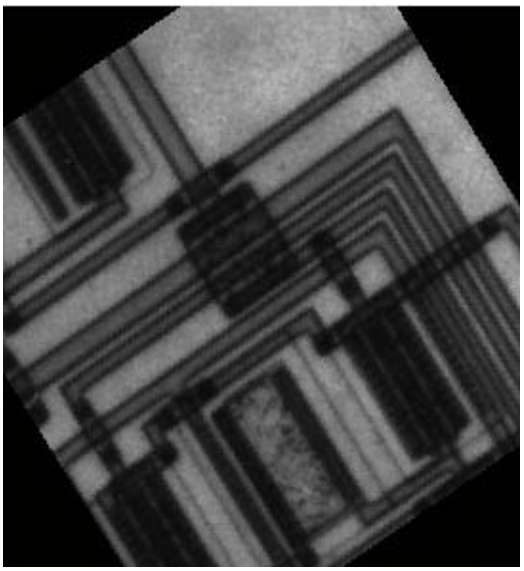


Figure 2.2 Original image

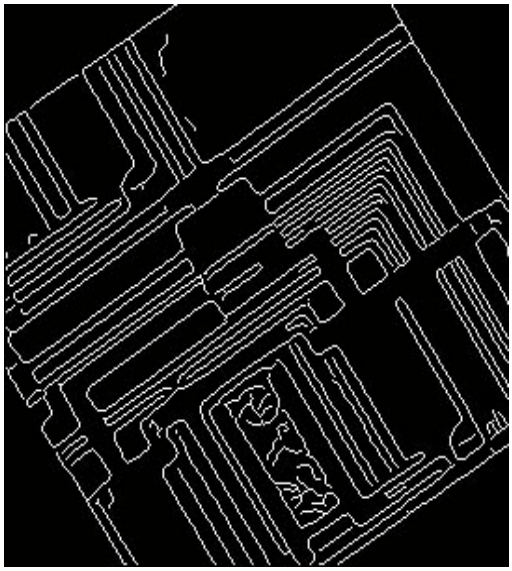


Figure 2.3 Edge detected image

2.3.2 Region Based Segmentation

Region growing is one of the simplest region-based image segmentation methods and it can also be classified a pixel-based image segmentation because it involves the selection of initial seed points.

The properties of region-based segmentation are described in the following:

- Partition an image R into sub-regions R_1, R_2,..., R_n (Figure 2.4)

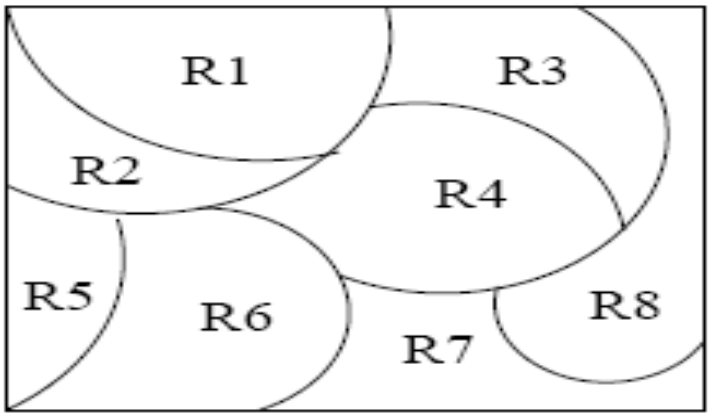


Figure 2.4 Sub-regions of R

Suppose $P(R_i)$ is a logical predicate, that is, a property that the pixel values of region R_i satisfy (e.g., the grey level values are between 100 and 120).

- The following properties must hold true:

(a) $\bigcup_{i=1}^{n} R_i = R$ (2.3)

(b) R_i is a connected region, i = 1, 2, ...,n

(c) $R_i \cap R_j = \emptyset \; for \; all \; i = 1,2, ...$ (2.4)

(d) $P(R_i)$ = TRUE for i = 1,2,...,n.

(e) $P(R_i \cup R_j) = FALSE \; for \; any \; adjecent \; R_i$ (2.5)

$P(R_i)$ is a logical predicate defined over the points in set $P(R_k)$ and $\varnothing$ is the null set.

(a) indicates that the segmentation must be complete; that is, every pixel must be in a region.

(b) requires that points in a region must be connected in some predefined sense.

(c) indicates that the regions must be disjoint.

(d) deals with the properties that must be satisfied by the pixels in a segmented region-for example R_i = TRUE if all pixels in R_i have the same grey level. The condition (e) indicates that region R_i and R_j are different in the sense of predicate P.

Region growing techniques are generally better in noisy image, where borders are extremely difficult to detect. Homogeneity is an important property of regions and is used as the main segmentation criterion in region growing, whose basic idea is to divide an image into zones of maximum homogeneity [Chang and Li 1994]. The criteria for homogeneity can be grey-level, colour, texture etc.

The most natural method of region growing is to begin the growth in the raw image data, each pixel representing a single region, and these regions will be merged according to the criteria with their neighbouring regions. Region splitting is the opposite of region merging, and begins with the whole image represented as a single region. Therefore, the existing image regions are sequentially split to satisfy the criteria. Even though the same criteria are used; region splitting does not result in same segmentation with region merging. Region splitting methods generally use similar criteria of homogeneity as region merging methods, and differ only in the direction of their application. A combination of splitting and merging may result in a method with the advantages of both approaches. Split and merge approaches work using pyramid image representations; regions are square shaped and correspond to elements of the appropriate pyramid level. If any region in any pyramid level is not homogeneous, it is split into four sub-regions; if four regions exist at any pyramid level with approximately the same value of homogeneity measure, they are merged into a single region in an upper pyramid level. The criteria plays a major role in split and merge algorithms [Chen et al 1991].

2.3.3 Level Set Methods

The basic idea of level-set methods is to represent the domain $\Omega(t)$ as the negative sublevel set of a function $\Phi(x,t)$,i.e.

$$\Omega(t) = \{x \mid \Phi(x,t) < 0\} \qquad (2.6)$$

This level set function $\Phi(x,t)$ is evolved via a special Hamilton-Jacobi equation, the so called level set equation：

$$\partial_t \Phi + V_n|\nabla\Phi| = 0 \qquad (2.7)$$

where V_n denotes the normal velocity, and it can be shown that for smooth shapes the evolution coincides with a standard motion of the shape with normal velocity V_n on its boundary. Figure 2.5 shows the basic sketch of level set method.

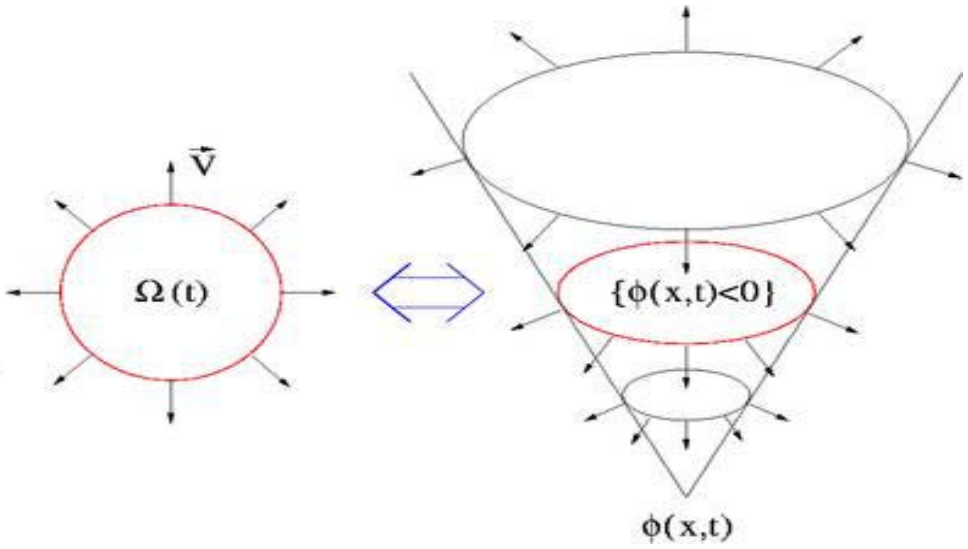


Figure 2.5 Sketch of level-set

In recent years, a large amount of work on image segmentation has been implemented using active contour methods [Kass et al 1987] [Chan and Vese 2001]. Active contours, or 'snakes', are dynamic curves that move within an image domain to capture the desired image features. These models are used extensively in image processing applications, including edge detection, shape modelling, segmentation and in general to detect object boundaries. There are two kinds of active contour models: parametric snakes and geometric snakes. Parametric snakes are represented explicitly as parameterized contours and the snake evolution is carried out on the predetermined spline control points only. Geometric snakes, on the other hand, are represented implicitly as the zero-level sets of higher dimensional surfaces, and the uploading is performed on the surface function within the entire image domain.

Geometric active contours were first introduced separately by Kass et al [Kass et al 1987] and Malladi et al [Malladi et al 1995]. These models are based on curve evolution theory [Kimia et al 1995] and a level set method [Osher and Fedkiw 2002]. This approach has two main advantages over the traditional parametric active contours. Firstly, the contours that were represented by the level set function can break or merge naturally during the evolution and the topological changes are thus automatically handled. Second, the level set function always remains a function on a fixed grid, which allows efficient numerical schemes. Early geometric active contour models [Malladi et al 1995] are based on Lagrangian formulation. The other method which is called the variational level method [Chan and Vese 2001] is discussed later. The variational level set method is more convenient and natural for incorporating additional information, such as region-based information [Chan and Vese 2001] and shape-prior information [Vemuri and Chen 2003], into energy functions that are directly formulated in the level set domain, and therefore produce more robust results.

2.3.4 Hierarchical Segmentation

Guigues' scale-sets hierarchical segmentation algorithm [Guigues et al 2006] is another important segmentation method because it makes both the segmentation criterion and the scale parameter explicit. The scale parameter becomes an output in multi-scale or hierarchical methods. It is then up to the post-segmentation stages to either select the most appropriate scale, or to analyse the results as a whole. This section briefly introduces Guigues' segmentation algorithm.

A partition P over a domain D is a division of D into separate pieces $pi \in 2^D$ (also named cells), as $P=\{p_1, p_2,\ldots\}$, such that $U_i P_i = D$ (the whole domain is covered) and for all i and j, $i \neq j \Rightarrow p_i \cap p_j = \emptyset$ (the cells are disjoint). We name part

D the set of all possible partitions of *D*. For an $x \in D$, let $P(x)$ denote the cell of *P* which contains *x*. A partition $P=\{p_i\}_i$ is coarser than another partition $Q=\{q_j\}_j$ (both over the same domain *D*), and we write $P \geq Q$, if, for all $x \in D$, $P(x) \ni Q(x)$. Conversely, *Q* is finer than *P*, $Q \leq P$.

Let's use as domain *D* a finite subset of Z^2, as the support for our image. Consider a partitioning algorithm A_I applied to an image *I*, whose result depends on a one dimensional parameter λ, the scale parameter.

$$\mathrm{A_I}: \mathrm{R} \rightarrow \mathrm{part(D)} \tag{2.8}$$

which is causal, that is,

$$\lambda_2 \geq \lambda_1 \Rightarrow \mathrm{A_I}(\lambda_2) \geq \mathrm{A_I}(\lambda_1) \tag{2.9}$$

So we can represent the partitions $A_I(\lambda)$, for all $\lambda \in R$, as a tree or hierarchy *H*. Because A_I is causal, the set $\Lambda(c)$ of values λ for which a given cell *c* is found in the partition $A_I(\lambda)$ is an interval $\Lambda(c)=[\lambda^+(c), \lambda^-(c)]$ (and $\lambda \in \Lambda(c) \Leftrightarrow c \in A_I(\lambda)$). We call $\lambda^+(c)$ the scale of appearance of *c*.

For each λ, Guigues' algorithm efficiently finds the partition that optimizes the balance between goodness of fit to the data and complexity. Theories of model inference (variational [Mumford and Shah 1989], Bayesian [Geman and Geman 1984] or minimum encoding [Leclerc 1989]) indicate the need for a regularization that takes into account the size or complexity of the solution. Usually, the end goal is finding a partition $P=A_I(\lambda)$ which minimizes an energy such as

$$\mathrm{E(P,\lambda)} = \mathrm{D(P)} + \lambda \mathrm{C(P)} \tag{2.10}$$

where $D(P)$ measures how badly *P* represents the original image and $C(P)$ measures the complexity of *P*. To avoid a problem known as “spill-over”, in which all the edges of an interesting region are in the hierarchy *H*, but the region itself is not a cell in *H* (i.e., it is not found as a single piece there), we consider,

instead of this hierarchy, a flattened version, a graph *G* consisting of the finest (lowest) level of the hierarchy, with edges attributed with the maximum level (λ) they reach in the hierarchy [Trias-Sanz 2005].

2.4 Methods of 3D Environment Reconstruction

Many 3D environment reconstruction systems have been developed due to their wide and useful applications such as: flood forecasting, forest fire monitoring, geographical information systems and flight simulations. Over the past few years, it has become increasingly common to use image-based reconstruction techniques on realistic terrain models [Sequeira et al 2001] [Debevec 2003] [Hakim 2002]. A standard approach to creating a 3D model is to build it from scratch using tools such as CAD software, which offers roads, vegetation and building blocks in the form of primitive 3D shapes. Some survey data or measurements from drawings and maps are also necessary. However, this geometry-based modelling technique is time consuming, impractical, and costly for large-scale projects. Although many applications apply this approach—even TV programs in Europe use it to render sites that no longer exist—the created models look computer-generated rather than realistic. They also generally don't include fine details or irregular and sculpted surfaces. Several recent techniques [Shum and Kang 2000] aim to increase the level of automation and realism by starting with actual images of the object or directly digitizing it with a laser scanner.

2.4.1 Related Work in Urban Modelling

One approach to modelling existing cities is the use of GIS data (building footprints) and aerial imagery [Takase et al 2003] [Ftacnik et al 2004] [Scholze et

al 2002]. These systems rely on highly calibrated aerial images as the main input for height determination and roof generation of buildings. With those systems, impressive results have been generated, but they are unsuited for the reconstruction of ancient ruined cities, and the textures on the facades, where still available, have to be added manually or through mobile mapping procedures.

In architecture, shape grammars [Stiny and Gips 1971] were successfully used for the analysis and construction of architectural designs [Stiny and Mitchell 1978] [Koning and Eizenberg 1981] [Flemming 1987] [Duarte 2002]. Shape grammars use production rules, which are defined directly on shapes (labelled points and lines). These rules iteratively generate a design by creating more and more detail. The derivation is usually done manually, or by computer with a human supervising the rule application.

Wonka et al [Wonka et al 2003] successfully introduced shape grammars to the computer graphics community by making them more amenable to computer implementation. In their approach, building designs are derived using a parametric set grammar. Their production rules consist of geo-metric split operations, which hierarchically subdivide the façade structure. They generated buildings with rich geometric detail for several different architectural styles.

2.4.2 Road Extraction

Extraction of curvilinear features, especially roads, has been a popular research topic in the computer vision and remote sensing communities. During the last three decades, many approaches for road extraction have been developed in the literature. Most of them are quite different in nature due to the differences in available data sources, assumptions about road models, and human interaction [Laptev 1997][Park and Saleh 2001]. These methods usually assume relatively

simplistic road models and make little use of a priori knowledge. They are thus sensitive to disturbances like shadows, occlusions and varying resolutions. Park and Saleh [Park and Saleh 2001] made a comprehensive survey for automated and semi-automated extraction techniques of linear features from panchromatic and multispectral aerial and satellite imagery. The techniques are evaluated with respect to methodology, strengths, drawbacks, and implementation approach. The viability of hyper-spectral data is extrapolated for same purpose of utilization. The most important factors influencing the nature of a road extraction approach are the data sources available and the need of an operator to provide control information [Laptev 1997].

The existing road extraction approaches utilize a wide variety of data sources largely to eliminate the uncertainty of road classification. The main data sources include aerial and satellite images with different resolution and spectral characteristics, and DTMs sometimes. Some automatic methods use existing GIS data as cues to improve extraction accuracy and reliability or to detect changes for updating [Zhang et al 2001]. Contextual information is taken into account to guide the extraction of roads [Ruskone 1996]. Much of the work is concentrated on low resolution (a few pixels in width), primarily rural areas rather than urban areas due to the complexity of urban scenes [Price 1999] [Hinz and Baumgartner 2000].

As has been demonstrated in prior research, the reflectance or spectral response as the only resource for road extraction is not sufficient, especially in urban areas. The limiting factors are reflectance depending on viewing conditions and illumination, variability of the surface materials occlusion and resolution [Price 1999]. The similarity in reflectance between objects leads to a high rate of misclassification. For example, in aerial and high-resolution satellite imagery, the

roads are hardly distinguishable clearly from buildings when they are coated with same materials and thus similar grey patterns are the present. The situation becomes worse when roofs or walls of tall buildings occlude or cast a shadow on the streets in built-up urban areas where neither automatic nor semi-automatic road detection methods could work reliably. In addition, most existing approaches work in 2D images, thus neglecting valuable information inherent in 3D processing when using elevation data.

A successful strategy should use specific techniques and fuse most fruitful data to handle different tasks with different road types and contextual complexity in the scene. Using a DSM, the occluded or shadowed areas aforementioned can be derived with the sun position known. This will provide useful information to explain and then bridge the gaps between broken road segments. Zhang et al [Zhang et al 2001] used height information derived by subtracting the DTM from the DSM to reason if a region is on the ground and to compensate the missing information in classification of aerial colour images. With increasing availability of Light Detection and Ranging (LIDAR) data, exploiting LIDAR elevation information and imagery for road extraction has also been initially investigated. The LIDAR data is found to be able to improve the analysis of high-resolution image data for detecting buildings and roads especially in urban areas. In cases when shadows cover objects of interest, their shape can be well described due to height information [Hofmann 2001].

LIDAR intensity data has good separability if the wavelength of the laser used is suitable for ground materials. Song et al [Song et al 2002] calculated a transformed divergence measure for intensity data to compare relative separabilities between object classes including asphalt road, grass, building roof

and tree. In particular, the separabilities for asphalt road vs. grass and asphalt vs. tree are rather high. By fusing elevation information, the intensity data is a valuable data source for road classification [Hu and Tao 2003]. The use of multi-return range data can ease the deletion of penetrable vegetation. Alharthy and Bethel [Alharthy and Bethel 2003] developed a similar algorithm to detect roads in urban areas using the intensity data and range data. However, the intensity data is usually very noisy due to the continuously varying reflection angles. Some materials appear with different reflectivity as the inclination angle and thus the reflection angle changes. To better represent the material's reflectance characteristics, the intensity values have to be normalized by the laser reflection angles [Song et al 2002]. However, this is not practical for end users because the normalization requires knowing the orientation information at the time of acquisition.

Another major factor influencing road extraction is the requirement of interaction between the algorithm and an operator. The semi-automatic strategy requires a human operator to input seed points interactively and optional information such as road width etc. to guide the extraction procedure. The optimal paths between seed points are then found by profile or template matching [Airault et al 1996] [Vosselman and de Gunst 1997], multi-resolution approach [Couloigner and Ranchin 2000], snakes or energy minimizing approaches [Laptev 1997][Gruen and Li 1997], dynamic programming [Fischler et al 1981][Gruen and Li 1994], cooperative algorithms [McKeown and Denlinger 1988], or Kalman filtering [Vosselman and de Knecht 1995]. The snakes or energy minimizing approaches work by defining appropriate energy functions based on radiometric and geometric assumptions for features of interest. They refine the solution to

minimize the energy function beginning with an initial estimate of the feature. To extract meaningful features successfully, template matching technique requires that the features possess similar grey patterns. Additional geometry constraints are often utilized to guide the matching. If more than one image is used, this can be done in 3D [Gruen and Li 1997].

Great effort has been made to increase the automation of road extraction. The automatic strategy finds candidate road segments by edge or line detection methods, and then tracks, verifies and links accepted segments by perceptional organization methods [Wang and Trinder 2000][Hu and Tao 2003b] or knowledge based methods [Stilla and Hajdu 1994] [Trinder et al 1997] to form a complete road network. Although the full automation is not likely to be reliable in many cases, it is the ultimate goal. Some semi-automatic approaches can be extended to fully automatic ones by means of automatic seed point detection [Zlotnick and Carnine 1993] [Mayer et al 1997].

The roads that are extracted automatically may have a low qualified representation at various aspects, and should be refined. In Vestri and Devernay [Vestri and Devernay 2001], all the junctions are processed at the same time. For each type of junction, they randomly sample two points in the different branches, estimate the position of the junction and compute residuals for all random sets. Then the solution that minimizes the median of residuals is selected. Wiedemann [Wiedemann 2002] presented an approach for improving the extracted road crossings as well as a scheme for the quality evaluation of the results. The topology reconstruction detects and removes all cycles within the crossing areas. The geometry reconstruction examines all combinations of main and branching roads, which are possible according to the given road segments and typical road

crossing models. The combination of main and branching roads that reaches the best score is selected as the final road crossing.

Many road models have been developed utilizing some basic road characteristics. Road characteristics can be classified into five groups: geometric, radiometric, topological, functional and contextual properties [Garnesson et al 1990]. Example characteristics are large length, small width variance, directional consistency, symmetry, homogeneous grey levels, good contrasts with both sides, intersecting each other, and so on [Vosselman and de Knecht 1995] [Barzohar and Cooper 1996] [Fischler and Heller 1998]. Different combinations of the above characteristics should be used to model roads in different resolution levels and contexts [Baumgartner et al 1997] [Mayer et al 1997] [Price 1999] [Wang and Trinder 2000]. Hinz and Baumgartner's [Hinz and Baumgartner 2000] strategy is based on a detailed road model including lanes, road markings and their context using both image and DSM information so that the inherently high complexity of urban scenes is reduced.

In high-resolution imagery, a road is often modelled as a continuous and homogeneous ribbon, which has internal structures such as lane markings and roadsides. Existing experiences show that the use of road models and varying strategies for different types of scenes are promising. The fusion of centrelines from low resolution and roadside edges from high resolution (<1 m) has proven to be very helpful in inferring more reliable results [Steger et al 1995] [Trinder and Wang 1998]. In Couloigner and Ranchin [Couloigner and Ranchin 2000], a novel multi-resolution approach is developed for semi-automatic extraction of streets. The accurate positions of roadsides and central reservations are determined through the analysis of multi-level wavelet coefficients. Instead of utilizing the

above characteristics describing individual road segments, local or global geometric and topological constraints can discover grid layouts of road networks. Price [Price 1999] designed a feature-based hypothesis and verification paradigm to find urban street grids from single or multiple aerial images by applying a local grid constraint. The method assumes that roads have visible edges without significant occlusions. An initial seed intersection that provides the size and orientation of the regular grid is selected manually. Road crossings are junctions of individual road segments of approximately constant width and height. Then the grid is iteratively expanded by adding new units. "In each iteration, the new segments are refined and evaluated by simultaneously matching their sides to image edges. Thus longer portions of the roadsides must be visible at least in one of different overlapping images" [Price 1999]. During final verification, height information and contextual knowledge are used to adjust the positions of several consecutive road segments and to remove short portions. By applying a global grid constraint, Hu and Tao [Hu and Tao 2003] proposed a procedure to reconstruct the street grids of the 3D grid road network in urban areas from LIDAR data also utilizing a road segment based hypothesis and verification paradigm.

2.4.3 Building Extraction

Buildings are the most important artificial objects consisting of piece-wise and continuous surface segments over the ground. Different data sources have been used for the building extraction task. These data sources include aerial and satellite images, 2D building ground plans from GIS databases, and high-quality DSMs. The task of building extraction is to determine building locations, ground elevation, orientations, size, rooftop heights, etc. Most buildings can be described

to sufficient details in terms of general polyhedra, i.e., their boundaries can be represented by a set of planar surfaces and straight lines. Further processing such as expressing building footprints as polygons is preferable for storing in GIS databases.

Like most feature extraction tasks, building extraction can be implemented in either semiautomatic or automatic strategies, and data-driven and model-driven techniques are commonly used. Some algorithms process the raw LIDAR point clouds directly or grid-based images converted from LIDAR data; others algorithms use these two data structures at different processing stages.

The semi-automatic building extraction approaches often prepare a set of building primitives for typical house types and roof shapes. Human operators place appropriate primitives and combine them to model complex structures. Various automated methods could assist operators in measuring and refining 3-D wire frame models. Brenner [Brenner 1999] reviewed several interactive modelling tools for 3D building reconstruction including ObEx, the so-called Stuttgart approach and CC-Modeller. The former two measure building primitives using several (monoscopic view) aerial images or LIDAR data and 2D ground plans. The latter uses stereo measurement of points in aerial images and uses two steps. In the first step, a structured point cloud containing all eaves and ridge points is obtained using strictly manual point measurement. The second step is automatic and consists of grouping points into planar faces and the generation of roof and wall faces.

In Rottensteiner [Rottensteiner 2001], a method for semi-automatic building extraction together with a concept for storing building models alongside with terrain and other topographic data in a topographical information system is

presented. His approach is based on the integration of building parameter estimation into the photogrammetric process applying a hybrid modelling scheme. A building is decomposed into a set of simple primitives that are reconstructed individually and are then combined by Boolean operators. The internal data structure of both the primitives and the compound building models is based on the boundary representation methods.

The automatic building extraction methods can be fulfilled by two sub-procedures, i.e., building detection and building reconstruction [Weidner 1995], which may not be clearly distinguishable. Full automation of building reconstruction is not yet reliable enough for practical production in most cases due to the great complexity of building architecture. Automatic extraction of buildings would go a long way to making more wide use of available geo-spatial data sources possible such as ground plans.

Some researchers have focused on the automatic extraction solely or mainly based on DSMs. The methods using only LIDAR data mainly utilize the geometric properties of buildings, and have to go through the building detection and reconstruction steps. In Weidner and Forstner [Weidner and Forstner 1995], the DSM is computed using image and feature pyramids and the final surface is then refined by means of local adaptive regularization techniques. Building detection is based on the fact that buildings are higher than the topographic surface, which is estimated using mathematical morphology on the DSM. The window size requires a priori knowledge of the maximum building size in the scene. Buildings are reconstructed depending on their complexity. Two kinds of parametric models are used for simple buildings with either a flat or a symmetric sloped roof. Prismatic models are used for complex or connected buildings.

In Maas and Vosselman [Maas and Vosselman 1999], two techniques for the determination of building models are developed. Based on the analysis of invariant moments of point clouds, closed solutions for the parameters of a standard gable roof building model with a rectangular ground plan are derived from 0_{th}, 1_{st} and 2_{nd} order moments. Asymmetric deviations like dorms on roofs can be modelled too. Heterogeneity in the point distribution will lead to biased parameters. Models of more complex buildings are determined using a data driven technique based on intersecting planes that are fitted to triangulated point clouds. Most problems occur in the determination of the outline of the building, especially when trees are near the building. Vosselman [Vosselman 1999] presented an approach based on the detection and outlining of planar faces. The planes of the faces are determined by clustering points. The outlines are determined by a connected component analysis assuming that all the edges are either parallel or perpendicular to the main building orientation.

Building surfaces including roofs and walls can be roughly approximated by constructing a TIN for points composing a building. The TINs of 3D points sampled on object surfaces and the simplification and refinement methods have been extensively studied for approximating object surfaces [Heckbert and Garland 1997]. The simplification method is a fine-to-coarse approach, and starts with an exact fit, and creates approximations with less and less details [Wang and Schenk 2000] while the refinement method is a coarseto-fine approach and starts with a minimal approximation, and generates more and more accurate ones [Hu et al 2003]. Most existing roof reconstruction methods are based on the simplification concept. They first try to aggregate the points that possibly belong to separate patches of a complex roof, and a plane fitting is then performed to get parameter

values for each planar patch. The plane detection methods reported in literature include clustering of triangles, 3D Hough transformation, and clustering of 3D points with or without using ground plans [Maas and Vosselman 1999] [Vosselman and Dijkman 2001] [Gamba and Houshmand 2002]. Those adjacent triangles with close normal directions may be grouped, and a plane equation is fitted to the vertices of the grouped triangles. Then the fine wire-frame model of a building's surface can be obtained by calculating and organizing the intersection lines between planes.

Wang and Schenk [Wang and Schenk 2000] proposed an edge based building detection and a TIN-based building reconstruction. The building models are reconstructed by triangulating each cluster of identified building points and grouping those fragmentary triangles into piecewise planes. Finally, the tri-intersections of those average planes are used to derive building corners and their relative orientations. The study area has nine buildings ranging from large to median to small sizes [Wang and Schenk 2000]. All the buildings in the area have rectangular or near rectangular shape with peaked roof, flat roof and multiple level flat roof, respectively. The range data plays a major role in the building extraction; although it may not be enough to complete the processing in many cases using only range data. The combination of geometric measures is proven to be effective in classifying buildings.

The 3D Hough transformation is used to extract planar faces in Vosselman and Dijkman [Vosselman and Dijkman 2001]. However, plane extraction with 3D HT has a high computational complexity determined by the number of 3D points and the size of a discretization of the two angles. In Elaksher and Bethel [Elaksher and Bethel 2002], the parameter space of the 3D Hough transformation is reduced

from three to two by using a simplified model assuming that only one of the roof slopes is not zero along or perpendicular to the main building orientation.

Some methods focus on the building detection stage. The derived building footprints or polygons can be used to update the building's basemap that is an integral part in spatial databases of GIS. Baltsavias et al [Baltsavias et al 1995] used an edge operator, mathematical morphology, and height bins for the detection of objects higher than the surrounding topographic surfaces. Hug [Hug 1997] showed the detection and segmentation of houses from ScaLARS height and intensity data based on morphological filtering with successive progressive local histogram analysis; in addition, they use the laser reflectivity measure for discerning manmade objects from vegetation via binary classifications. In Hu et al [Hu et al 2003], several algorithms, including the constrained searching in Hough space, enhanced Hough transformation and sequential linking technique are developed to reliably express building footprints as rectangles, quadrangles or polygons. These algorithms utilize the hypothesis verification paradigm and a final refinement.

The active contour model is used to locate the boundary of a building in Yoon et al [Yoon et al 1999]. Hu and Tao [Hu and Tao 2002] also tested the active contour model to locate building boundaries using greedy algorithm, and found several drawbacks, including the requirement of initial boundary position, possible shift to neighbouring buildings during iterations and the difficulty of reconstructing regular shapes using a priori knowledge (e.g., orthogonality). Research effort has been made to rectify building boundaries that are detected automatically. Sester [Sester 2000] presented solutions for generalization problems using least-squares adjustment, focusing on the scale dependent

representations. In Vestri and Devernay [Vestri and Devernay 2001], angle constraints are applied to refine the corners and junctions of polygon models, and the problem is solved by optimising an objective function to preserve the global consistency.

An important task in building detection is to distinguish between buildings and vegetation. The discrimination between buildings and vegetation based on a Bayesian nets classification algorithm using local geometric properties is discussed in Brunn and Weidner [Brunn and Weidner 1998]. Zhan et al [Zhan et al 2002a] applied an object-based classification to detect building footprints. First, the image segments belonging to the building class are identified by the vertical wall analysis, which is to examine the sliced LIDAR DSMs; then colour infrared imagery is used to calculate the NDVI, which is then assimilated to refine the identified building segments [Zhan et al 2002b].

Rottensteiner [Rottensteiner 2001] found that roof structures could be extracted as a first step toward the reconstruction of polyhedral building descriptions. The use of shape measures based on differential geometric properties of range data for building detection is not sufficient and also poses high requirements on the resolution and quality of the data. In this situation, multiple returns and LIDAR intensity data could help much. Elberink and Maas [Elberink and Maas 2000] presented their work on segmenting 0.5-m to 1-m LIDAR data in an unsupervised *k* means classification of objects (e.g., buildings and trees) using anisotropic height texture measures. The texture is defined quantitatively and qualitatively by height, variation of height in local windows and measures such as contrast and homogeneity. An important aspect of the work is the analysis of the benefit of intensity data as well as first and last returns range data. In Hu et al [Hu

et al 2003], the first and last returns LIDAR range data are used to produce the VSM that represents penetrable high vegetation objects such as trees; then the binary objects in the up segmented DNM that have non-empty intersections with the VSM are deleted with only buildings retained.

The methods using both DSMs and auxiliary data such as GIS maps and multi-spectral images could overcome the drawbacks of specific imaging sensors. The integration of LIDAR data and ground plans is shown to be successful, and detailed reconstruction of buildings can be obtained automatically even for LIDAR data with relatively low point densities. This type of approach has the advantage of skipping the building detection step with ground plans known and focusing on building reconstruction stage. Henricson et al [Henricson et al 1996] used information from CIR images to separate elevation blobs detected in a DSM from stereo image matching into the classes of buildings and trees. Lemmens et al [Lemmens et al 1997] showed the fusion of LIDAR altimeter data with 2-D digital maps in a topographical database to derive heights for roofless cube type building primitives.

Haala and Anders [Haala and Anders 1997] demonstrated two approaches aiming on the combination of DSMs, aerial images and ground plans for the reconstruction of 3D buildings. The ground plans provide very precise information about the building outlines. The first approach extracts breaklines from both DSMs and image data. Then the breaklines of high reliability are combined with the grey value edges of high geometric accuracy to reconstruct a rather simple type of buildings. The second approach extracts planar surfaces likely to be roof planes from the DSMs and uses polyhedron as the building model to reconstruct very general type of buildings. It utilizes given ground plans as a

priori information and extracts planar surfaces likely to be roof planes from DSMs.

Haala et al [Haala et al 1998] derived parameters for 3D CAD models of basic building primitives by least-squares adjustment minimizing the distance between a laser DSM and corresponding points on a building primitive. The building boundaries were derived from ground plans. The implementation was limited to four standard building primitives and their combinations. Further refinement has to be performed interactively. Haala and Brenner [Haala and Brenner 1999] combined multi-spectral information provided by an aerial CIR image with geometric information from a LiDAR DSM. A pixel-based classification is applied for the extraction of buildings, trees and grass-covered areas, whereby the normalized DSM is used as an additional channel in combination with three spectral bands. Additional constraints are obtained for reconstruction by using the assumption that the given ground plan is correct and exactly defines the borders of the roof.

Brenner [Brenner 2000] also presented the reconstruction of building of complex structures using LIDAR DSMs and ground plans. Roof surface primitives are segmented, and a rule-based approach decides which segments can be explained by the chosen building model. Finally, the roof is built from the primitives that have been accepted, closing gaps that are caused by the deletion of unexplainable regions. In Ameri [Ameri 2000], building models are firstly created based on DSMs, and are then verified by back-projecting them to images. Matching the model edges with image edges, the accuracy of the model parameters can be increased especially with respect to the building outlines.

The quality evaluation for building detection results has been initially addressed. Geibel and Stilla [Geibel and Stilla 2000] compared four segmentation procedures using LiDAR data with a density of four points/m^2, and proposed an evaluation function to estimate the segmentation quality of a complete scene. The evaluation function subjectively weights measures for the over-segmentation and under-segmentation. However, at the present there is a lack of objective measures concerning reconstruction times, success rates and reconstruction quality, especially when a human operator is the part of a semiautomatic system and different ancillary data are used [Brenner 1999][Rottensteiner 2001][Lemmens et al 1997] [Vosselman and Dijkman 2001].

2.4.4 Classification of Vegetation and Man-made Structures

Filin [Sagi 2002] has proposed a surface clustering technique for identifying regions in LiDAR data that exhibit homogeneity in a certain feature space. His method computes a 7-tuple feature vector consisting position, tangent plane and relative height difference attributes for every point. The surfaces are categorized as high vegetation that exhibit rapid variations in slopes and height jumps, low vegetation, smooth surfaces and planar surfaces. The algorithm identifies these surface clusters iteratively by proposing a certain surface type for a region and identifying the points associated with that category within that region. The method is suitable only for broadly classifying vegetation, and terrain classes.

Song et al [Song et al 2002] developed a metric for determining the separability of different classes in LiDAR data based on return signal intensity of LiDAR points. They have shown that the return signal strength can be used for data classification purposes. In his work, the set of classes were designed based on prior knowledge of the target region and typical reflectivity of surfaces.

Andersen et al [Andersen et al 2002] present a bayesian recognition method for identifying individual tree crowns in LiDAR data for forested areas. Prior distributions are based on modelling of spatial distribution of foliage in the region and the tree position inferences are based on posteriori probability estimates. They have compared their results with photogrammetric measurements.

Some early work in LiDAR data classification has been carried by Axelsson [Axelsson 1999]. He has presented algorithms for filtering and classification of data points into terrain, buildings and electrical power lines. The method uses curvature based minimum description length criterion for classification.

Haala and Brenner [Haala and Brenner 1999] have presented a method for classification of buildings, trees and grass covered areas. They have used multispectral imagery in addition to LiDAR data for classification. Their classification is based on clustering pixels in a feature space derived from the elevation and spectral information.

2.5 Methods of 2D and 3D Flood Simulation and Prevention

During the past four decades, natural hazards such as earthquakes, droughts, floods, forecast fires, typhoon and volcanic eruptions have caused huge loss of human lives, the destruction of economy and social infrastructure, as well as environment damage. In fact, of all natural risks, floods pose the most widely distributed risk to life today. Every summer in the middle and lower reaches of the Yangtze River in China, people are threatened by the terrible floods and thousands of houses have been destroyed. Many countries spend a lot of money and material resources on constructing flood defence system, but how to achieve better practical prevention and decrease the expenditure are important aspects of

concern. In the following, we will talk about several types of flood and then describe five main approaches for flood prevention.

2.5.1 Flood Categories

Floods are among the most devastating natural hazards in the world, claiming more lives and causing more property damage than any other natural phenomena. A flood can be defined as any relatively high water flow that overflows the natural or artificial banks in any portion of a river or stream. When a bank is overflowed, the water spreads over the flood plain and generally becomes a hazard to society. When extreme meteorological events occur in areas characterized by a high degree of urbanization, the flooding can be extensive, resulting in a great amount of damage and loss of lives. There are many forms of floods: River Floods, which are formed from winter and spring rains, coupled with snow melt, and torrential rains from decaying tropical storms and monsoons; Coastal Floods are generated by winds from intense off-shore storms and Tsunamis; Urban Floods, as urbanization increases runoff two to six times what would occur on natural terrain; Flash Floods can occur within minutes or hours of excessive rainfall, a dam or levee failure, or a sudden release of water.

2.5.2 Water Conservancy Method

Water conservancy means using artificial methods to stand against the flood. Building dams and reservoirs and reinforcing the embankments are the most important approaches. Dams reduce expenditure on other forms of flood mitigation and reparation of flood damage. Reservoirs have multi-purpose especially in the upstream area of the river. We can also divert flood into specific river braches. Ideally, the flood mitigation strategy adopted in particular

catchments should be an optimal integration of the most appropriate techniques to afford the required standard of protection, taking into account the hydrological, land use, settlement and environmental characteristics of the valley.

2.5.3 Remote Sensing Approach

Spatially, remote sensing [Wickland 1989] [Curran et al 1997] exceeds other methods of data collection as vast areas of the Earth's surface can be covered with the use of aerial photography or satellite imagery relatively quickly. It also offers the ability to study inaccessible areas and at substantially lower costs compared to extensive ground studies. Remotely sensed data can be accessed within a matter of hours or days from the time in which it was collected, increasing the possibilities and allowing for the expansion of temporal studies. Due to the vast variety of sensors and forms of data collected, a study area or problem can be investigated by the most appropriate tool available, as various sensors provide different temporal, spatial and spectral information. It is important to consider more recently developed active sensors, such as microwave and laser techniques as they have the ability to penetrate tree canopies and cloud cover. For example, passive sensors such as Landsat Thematic Mapper^{TM} are limited by atmospheric conditions, as well as the inability to detect standing water beneath obstructions such as forest canopies [Townsend and Walsh 1998]. Radar systems are capable of detecting sharp land-water boundaries, and are capable of higher spatial resolutions than passive systems under similar situations [Engman and Gurney 1991]. Radars can penetrate clouds, darkness and, at the longer wavelengths, tree canopies [Smith 1997]. The use of such sensors for the same or similar applications has been demonstrated in a number of studies. Lowry et al. [Lowry et al 1981] demonstrated the ability of the synthetic aperture radar (SAR) for

providing all-weather flood area delineation. Hess et al. [Hess et al 1990] reviewed the use of radar detection for floods beneath forest canopies and determined that it is a proven method for such investigations, where other sensors would undoubtedly fail.

Remote sensing is now becoming more and more popular in detecting and forecasting inundation [Curran et al 1997] [Smith 1997]. In 2006, Marco Gianinetto et al proposed a straightforward method for mapping the inundated areas and for estimating flood water volumes through the use of optical Landsat multispectral data integrated with topography data [Gianinetto et al 2006]. The methodology used relies on spectral-temporal principal component analysis, logical filtering with DEM data, image segmentation and flood intensity estimation. They develop a decisional support tool for the allocation of resources destined for the reconstruction of the inundated municipalities. In 2005, Horritt [Horritt 2005] presented a rationale for assessing both the accuracy and precision of uncertain predictions of flood extent by means of remotely sensed images of inundation patterns. They used two criteria to test for the calibration and validation of a simple cellular flow model on a reach of the river Severn, UK. The accuracy of model predictions is quantified through use of the reliability diagram and the associated root mean square error between uncertain predicted inundations and observed flooded proportion, and the precision measured through use of an entropy-like measure. The two measures they utilized give an insight into the differences between the calibration and validation processes and the effectiveness of data from different satellite sensors in constraining uncertain model parameters.

2.5.4 Simulation of Flood Inundation

The simulation model can be used to predict flood wave propagation and provide the information about the area to be flooded, wave front arrival time and water depth etc. Therefore, it is a useful tool for establishing flood control and dam operating strategies as well as developing evacuation plans and warning systems for the areas having potential flood risk.

Hydrodynamic models play a major role in numerical simulation. The hydrodynamic models require several types of data as input, such as land use, soil type, soil moisture, stream/river base flow, rainfall amount/intensity, snow pack characterization, digital elevation model (DEM) data, and static data (such as drainage basin size). Model predictions of potential flood extent can help emergency managers develop contingency plans well in advance of an actual event to help facilitate a more efficient and effective response. Flood forecast can be issued over the areas in which remote sensing is complementary to direct precipitation and stream flow measurements.

Hydrodynamic modelling can be categorized into three main groups: 1) one dimensional, 2) two dimensional and 3) three dimensional. We will briefly discuss them in the following sections.

2.5.4.1 One-Dimensional Hydrodynamic Modelling

One-dimensional models are the simplest option available for modelling the flow conditions within a river channel. They are best suited for modelling the flow conditions within a river channel that is described by a set of stream cross sections. The simplest forms of one-dimensional models solve one-dimensional energy equations to compute water surface elevations at each cross section for steady

gradually varied flow conditions. These models can also accommodate momentum equations for rapidly varied flow conditions, such as a hydraulic jump. Slightly more sophisticated one-dimensional models simulate unsteady state flow conditions in river channels and solve cross-sectional averaged Saint-Venant equations to route the flow hydrograph and compute water surface elevation at each cross-section. For example, HEC-RAS [USACE 2002], a commonly used one-dimensional hydrodynamic model, has the capability to perform both steady and unsteady state simulations. A one-dimensional river flow model has been shown in Figure 2.6.

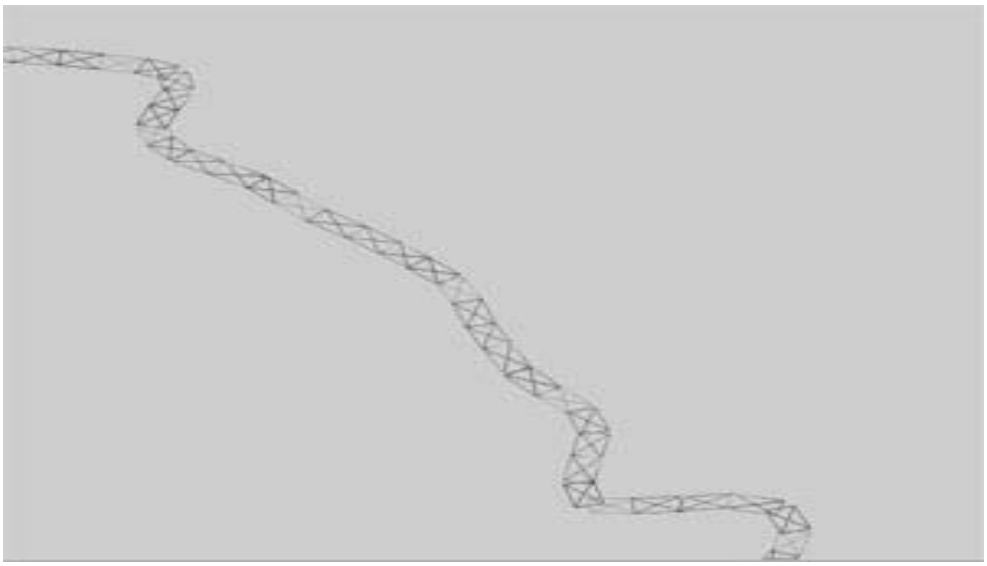


Figure 2.6 One-dimensional model

One-dimensional models have traditionally been used for flood plain mapping, and are still commonly used for this purpose. The model setup is easy and the computations are fast. However, they are limited by their ability to model two-dimensional characteristics such as channel meander, velocity currents, flooding in flat urban environments, lateral distribution of flows, etc. Some limitations of one-dimensional models are overcome by using two-dimensional models.

2.5.4.2 Two-Dimensional Hydrodynamic Modelling

One-dimensional models work in the longitudinal direction, and two -dimensional hydrodynamic models go one step further by adding the lateral component of river hydrodynamics to the system. Two-dimensional hydrodynamic models solve depth averaged mass and momentum equations to compute water surface elevations and velocities. In other words, at each point, three items are computed: water depth, and velocities in two directions (x and y). Most two-dimensional models operate under hydrostatic assumption, which means the accelerations in the vertical direction are negligible. The spatial or computational domain of the river system is divided into a set of elements, and for each element, the velocity vectors are assumed to point in the same direction over the entire depth of the water column at any instant of time. The computational domain of two-dimensional hydrodynamic models can be represented by using either (x, y) coordinates or channel oriented curvilinear orthogonal coordinates. In the curvilinear orthogonal coordinate system, the data is plotted with respect to the flow direction. The hydrodynamic models, however, take the data in (x, y) coordinates and the computations are performed in curvilinear orthogonal coordinates [Hodges and Imberger 2001].

The two-dimensional models use either the finite difference method (FDM) or the finite element method (FEM) to solve the energy and momentum equations. The choice of solution method depends on several factors such as the computational speed, model setup, and shape of computational domain. However, in most instances, finite element methods are usually preferred over finite difference methods due to their ability to model complex geometries in an efficient manner [Jennings 2003]. Commonly used two-dimensional

hydrodynamic models, such as RMA2 [Barbara et al 2003] and FESWMS [Froehlich 1992] are based on finite element technique. MIKE 21 [DHI 2001] from the Danish Hydraulic Institute is an example of a finite difference model. Typical data requirements for two-dimensional models consist of geometry data, boundary conditions, and calibration data.

In general, collecting the terrain data, interpolating the data, and developing a suitable finite element mesh are some of the crucial points related to two -dimensional hydrodynamic modelling.

The geometry data mainly consist of channel bathymetry in the form of finite element mesh specified by (x, y) coordinates at each mesh node. The collection of geometry data for two-dimensional models, however, demands more resources compared to one-dimensional models.

The most commonly used boundary conditions for two-dimensional models are downstream water surface elevation and upstream flow rate. Calibration data for two-dimensional models mainly include point measurements of depth and velocity. These measurements are compared with model estimates, and the parameters are adjusted to match the measured values. Additional inputs to two -dimensional models consist of eddy viscosity values and roughness coefficients for various substrate types.

Two-dimensional models have both advantages and disadvantages compared to one-dimensional models. Advantages include two-dimensional distributions of flow and velocity, and the ability to simulate complicated flow patterns. The main disadvantage of two-dimensional models is that they are not as powerful as one -dimensional models for simulating flow across control structures such as weirs, pumps, tidal gates, etc. This disadvantage of two-dimensional models can be

overcome by combining them with one-dimensional models to use the best of both models [Morten and Wium 2003]. The crucial factor while using two -dimensional models is an accurate description of channel bathymetry. The RMA2 references manual [Barbara et al 2003] suggests that geometry and study design are the most significant factors in the model application. Geometry and study design mainly include the description of meshes, which in turn depends on the channel bathymetry.

Attempts to address the issue of channel geometry with two-dimensional models have been made both by scientists and engineers. Some of these attempts include refining the mesh resolution to account for small scale topographic features, such as boulders and woody debris [Crowder and Diplas 2000] [Gilvear et al 1999]. All these studies conclude that higher resolution produces better model results. Although mesh resolution is important, there are some basic issues that need to be addressed with the channel bathymetry. According to French and Clifford [French and Clifford 2000], these issues are related to the quantity and quality of the terrain data and the tools that are used to interpolate these data onto computational meshes. Even though the channel bathymetry plays a significant role in the model output, the procedures by which the terrain data are used to interpolate the bathymetry onto mesh nodes are poorly documented. There are, however, exceptions such as Carter and Shankar [Carter and Shankar 1997] who suggest that kriging [Oliver and Webster 1990] algorithms work better for interpolating river channel bathymetry. Figure 2.7 is a two-dimensional surface flow model which has been used for simulating flood inundation.

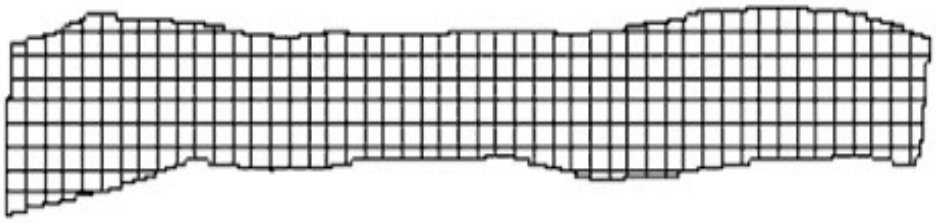


Figure 2.7 Two-dimensional hydrodynamic model

2.5.4.3 Three-Dimensional Hydrodynamic Modelling

Three-dimensional models are similar to two-dimensional models, except that the governing equations are not depth-averaged. Besides the two horizontal dimensions (x and y), the vertical dimension (z) is also modelled by introducing a number of layers in the water column. Advantages of three-dimensional models include their ability to model the vertical distribution of flow and velocity, which include stratification, diffusion and dispersion processes. Three-dimensional models, however, require additional computational time. Given the increase in computational speed of computers over past few years, computational time may not be an issue with three-dimensional models in the future.

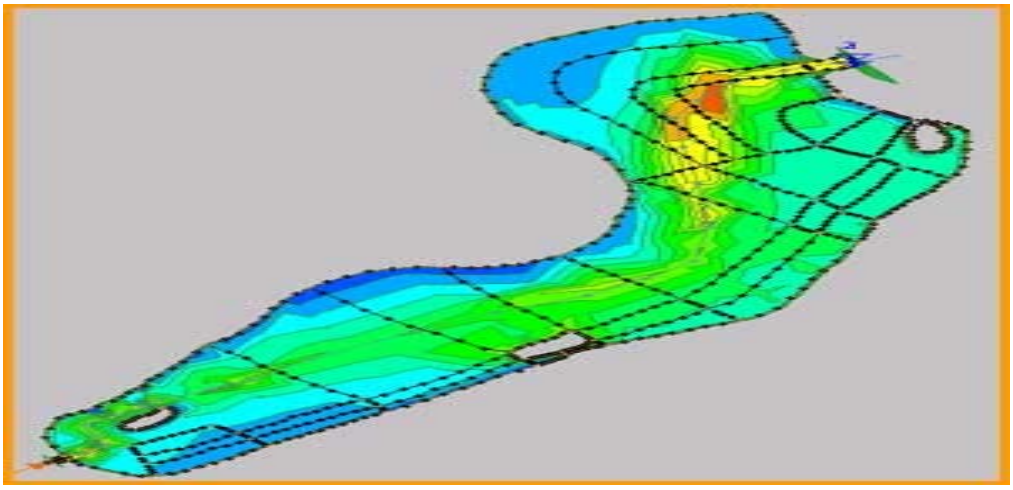


Figure 2.8 Three-dimensional river model

The data requirements for three-dimensional models are similar to two-dimensional models with additional measurements of velocity fields in vertical direction [David et al 2002]. The issues related to bathymetry data with two-dimensional models also apply to three-dimensional models. Figure 2.8 shows three-dimensional hydrodynamic model bathymetry grids for Charleston Slough.

2.5.4.4 Numerical Simulation Based on Hydrodynamic Model

Xin et al have described a 2D numerical model which is based on a finite volume method (FVM) [Ying et al 2003]. They simulated flood wave propagation due to a dam-break event. The intercell flux is computed by upwind method and water-level-gradient is evaluated by weighted average of both upwind and downwind gradient. Therefore, the conservation of mass and momentum is guaranteed and the scheme is easy to implement. The proposed scheme is extensively tested with various examples, including a partial dam-break problem, oblique hydraulic jump, and flood wave propagation in a river valley with natural topography. The numerical results are compared with analytical solutions and experimental data. However, this scheme generally obtains the accuracy and stability at the expense of increased algorithm complexity, which may result in increasing computational time. Dubicki and Stronska [Dubicki. and Stronska 2000] have proposed a physical based flood forecasting system on the upper and middle odra. They developed a timely and reliable forecasting system for the complicated river basin by means of MIKE 11 River modelling software and the flood watch forecasting system. They considered rainfall runoff modelling, hydrodynamic modelling and the implementation of flood watch for real time operation. They also kept the balance between accurate representation of the flood wave movement and extent and the need for rapid forecasts, but their system lacks in

the support of a digital terrain model, and so it cannot be applied to flood inundation mapping.

2.5.5 Other Flood Models

2.5.5.1 HEC-RAS

HEC-RAS is a one-dimensional steady flow hydraulic model designed to aid hydraulic engineers in channel flow analysis and floodplain determination. The results of the model can be applied in floodplain management and flood prediction studies. If you recall from hydraulics, steady flow describes conditions in which depth and velocity at a given channel location do not change with time. Gradually varied flow is characterized by minor changes in water depth and velocity from cross-section to cross-section. The primary procedure used by HEC-RAS to compute water surface profiles assumes a steady, gradually varied flow scenario, and is called the direct step method. The basic computational procedure of HEC-RAS for steady flow is based on the solution of the one-dimensional energy equation. Energy losses are evaluated by friction and contraction or expansion. The momentum equation may be used in situations where the water surface profile is rapidly varied. These situations include hydraulic jumps, hydraulics of bridges, and evaluating profiles at river confluences. For unsteady flow, HEC-RAS solves the full, dynamic, Saint-Venant equation using an implicit, finite difference method.

HEC-RAS is equipped to model a network of channels, a dendritic system or a single river reach. Certain simplifications must be made in order to model some complex flow situations using the HEC-RAS one-dimensional approach. It is

capable of modelling subcritical, supercritical, and mixed flow regime flow along with the effects of bridges, culverts, weirs, and structures.

2.5.5.2 LISFLOOD-FP

LISFLOOD-FP is a raster-based inundation model specifically developed to take advantage of high resolution topographic data sets. Channel flow is handled using a 1D approach that is capable of capturing the downstream propagation of a floodwave and the response of flow to free surface slope, which can be described in terms of continuity and momentum equations.

Floodplain flows are similarly described in terms of continuity and momentum equations, discretized over a grid of square cells which allows the model to represent 2D dynamic flow fields on the floodplain. We assume that the flow between two cells is simply a function of the free surface height difference between those cells [Estrela and Quintas 1994].

The model predicts water depths in each grid cell at each time step, and hence can simulate the dynamic propagation of flood waves over fluvial, coastal and estuarine floodplains. It is a non-commercial, research code developed as part of an effort to improve our fundamental understanding of flood hydraulics, flood inundation prediction and flood risk assessment.

2.5.5.3 TELEMAC-2D

The TELEMAC-2D [Galland et al 1991] [Hervouet and Van Haren1996] model has been applied to fluvial flooding problems for a number of river reaches and events [Bates et al 1998]. The model solves the 2D shallow water (also known as Saint- Venant or depth averaged) equations of free surface flow.

The TELEMAC-2D model uses Galerkin's method, which is an unstructured mesh of triangular finite elements of weighted residuals to solve

shallow water equations. A Streamline-Upwind-Petrov-Galerkin (SUPG) technique is used for the advection of flow depth in the continuity equation to reduce the spurious spatial oscillations in depth that Galerkin's method is predisposed to, and the method of characteristics is used for the advection of velocity. The resulting linear system is solved using a gradient mean residual technique and efficient matrix assembly is ensured using element by element methods. The time development of the solutions is dealt with using an implicit finite difference scheme and the moving boundary nature of the problem is treated with a simple wetting and drying algorithm which eliminates spurious free surface slopes at the shoreline [Hervouet and Janin 1994].

2.5.6 Risk Assessment and Evacuation Management

The fourth method refers to risk assessment management, evacuation path selection and risk response simulation. In 2004, Tomoharu and Michiharu [Tomoharu and Michiharu 2004] brought forward two types of micro simulation tools for the performance-based design of a flood risk management: 1) a micro model based simulation system of flood evacuation, 2) an object-oriented simulation model of communication processes among disaster prevention agencies. The micro simulation model of flood evacuation is based on parameterization of people's attitudes to flood risk and recognition of danger during flooding. The second micro simulation tool can be used for expressing concentration of the path of information and unbalanced element task descriptions quantitatively. The development of the flood simulation model was an iterative process. Initially, an executable prototype model was implemented, using fictive data. The prototype was based on an existing cyclone model developed by

Ermolieva [Ermolieva 1997]. Some experiments were performed on the prototype model and made the basis for the sharp version of the model, see [Brouwers 2002]. Lisa has proposed two different flood management policy scenarios which are implemented and evaluated through analysis [Brouwers and Hansson 2002]. Within this project, a geographically explicit flood simulation tool has been designed and implemented. The model integrates aspects of the geographical, the hydrological, the economical, and the social. The problem of this method is the uncertain factor which can lead to failure.

2.5.7 GIS Based Flood Prevention

The fifth approach is about GIS based flood prevention. A geographic information system (GIS) is a computer system that stores, edits, analyzes and displays data from a geographic perspective. GIS could be divided into at least 3 categories through the experience of a developing disaster prevention system. The first type of GIS is for regional analysis and decision-making based on data integrating detailed geographical historical (temporal) information of a region to the maximum extent. This type of GIS can give solutions for federal or local government requirements. The second type of GIS is for general public use, for example, to be used instead of a paper map. Requirement of data is mainly positioned on geographic objects which are subsets of the first type of GIS data that provide service information mainly as attributes which are gathered by unspecified people and system. The third type of GIS is for special purposed GIS such as car navigation systems of which functional requirements are almost finalized. Compactness, efficiency, and cost are important. Now combination of non-Internet GIS and Internet/Web GIS related to realization of disaster mitigation and environment analysis is the main research and application aspect.

Yaolin Liu and Zuohua Miao [Liu and Miao 2005] discuss a new theory for construction of floods prevention information system integrated with geography information system (GIS). They applied a new map projection which manipulates the earthly surface to horizontal surface. They grouped datasets in two types: attributive data and spatial data. Attributive data is managed with traditional data base management system (DBSM). GIS is used to operate the spatial data. Although their method can solve some piratical problems, there are still two disadvantages of the system. One is the flood prevention system is constructed in 2D environments other than 3D real world. The other thing is that they forecast the flood based on the analysis of historical data. As we know, in order to make the result more accurate, real-time data is preferred. Alemseged et al [Alemseged and Rientjes 2005] have pointed out simulated topographic properties had a major effect on simulation results and topography is a major factor determining flood inundation patterns as they develop over time. In their method, a 2D module is activated for the simulation of water flow in river reaches as well as river-bank overflow and flow at flood plains. They categorized data requirements for floodplain modelling into data input for analysis, calibration and verification, but flood simulation characteristics are affected by inundation extent, flow velocity, flow depth and flow patterns across the model domain. It has however not become clear from their study what generic aspect of the applied flood model approach causes the significant differences. Masaya et al [Masaya and Tesuhiko 2005] applied a diffusion equation model into their flood simulation. They implemented flood simulation in an Integrated Earthquake Disaster Simulation System (IDSS). In the IDSS, a simulated area can be divided into small regions where distributed kernels are responsible for local simulations. By using GIS data, the simulated

area is generated. The limitation of this project is only small area simulation. Also there are some errors in the flood acreage.

2.6 Conclusion

This chapter has reviewed several methodologies correlated with our research work which are methods of digital terrain model generation, methods of image segmentation and recognition, methods of 3D environmental reconstruction and methods of 2D and 3D flood simulation. Although many of the methods are feasible, there are still some limitations that could be improved. In the following chapters, we will propose our novel methods for digital terrain modelling, 3D environmental reconstruction and flooding models.

Chapter 3 Digital Terrain Model

Digital terrain generation has become a subject of significant interest in the last two decades. Terrain modelling algorithm and GIS data are basic aspects of the terrain generation. Modelling terrain is the initial step to construct our flood hazard warning system.

In this chapter, we propose the NTSP algorithm and pyramidal data arrangement structure for digital terrain generation. The main contributions of our terrain modelling work are listed below:

- It is able to perform an out-of-core rendering and to deal with massive GIS terrain data which is easy to be implemented using hardware memory.
- It is capable of processing and presenting the detailed terrain information with high resolution data set through in-core pyramidal data arrangement structure.

3.1 Introduction

A Digital Terrain Model (DTM) is a digital file providing a highly detailed representation of the topographical variations in the Earth's surface. Combined with other digital data, such as maps or orthophotographs, it can provide a 3D image of the land surface. Consisting of terrain elevations for ground positions at regularly spaced horizontal intervals, the added dimension and visualization offered by a DTM can help in many decision-making processes, urban planning and large scale flood warning.

The focus of most previous related works has been on the development of

algorithms for visualization of large scenes, notably based on a level-of-detail approach. Level of detail modelling is an approach to high-performance visualization that uses the concept of "levels of detail" (LOD): several descriptions of the objects in the scene are provided or automatically computed, with different levels of complexity [Rossignac and Borrel 1993][Thomas and Carlo 1993]. One of these representations is dynamically selected for rendering based on the viewing conditions and other factors. For example, at large scales, geometric models are necessary. At intermediate scales, texture mapping and similar techniques may suffice. However, this method is slow to support dynamic terrain.

In order to solve this problem, we propose the Nona Tree Space Partitions (NTSP) and pyramidal data arrangement structure for digital terrain modelling which has two main advantages:

- This partition scheme is able to perform an out-of-core rendering and to deal with massive GIS terrain data. As we only need to use nines terrain nodes for world visualisation for navagation, a whole world can be visualized as long as the necessary GIS data available.
- An in-core process with a pyramidal data arrangement structure is efficent for processing high resilution terrain with detailed image and interpolated elevation data.

This chapter is structured as follows: Section 1 introduces the research on digital terrain modelling; Section 2 describes 3D real world generation method from achieved digital data; Section 3 explains our proposed space partition for data storage and processing; Section 4 proposes our Nona Tree Space Partitions (NTSP) for DTM Modelling by using satellite images and real elevation data,

pyramidal data arrangement structure and new map projections; Section 5 draws the conclusion.

3.2 Generating 3D Real World

3.2.1 Real Data Calibration and Fusion

SRTM is a powerful near-global digital elevation model (DEM).This image leads to output accurate data sets with 1 arcsec (or 30m) horizontal resolution for the United States, and 3 arcsec (90m) for the rest of the world. Both distributions have a guaranteed vertical precision of 10m.

Because the original images are unedited, they contain occasional voids, or gaps, where the terrain lay in the radar beam's shadow or in areas of extremely low radar backscatter, such as sea, dams, lakes and virtually any water covered surface that are flat but don't look flat on SRTM tiles. Voids are points of the SRTM data tiles that contain no data; this happens to bad radar scatter and bad weather, the voids clearly disturb the terrain perception. As a beginning, we must rectify the data.

Figure 3.1 Original hgt data

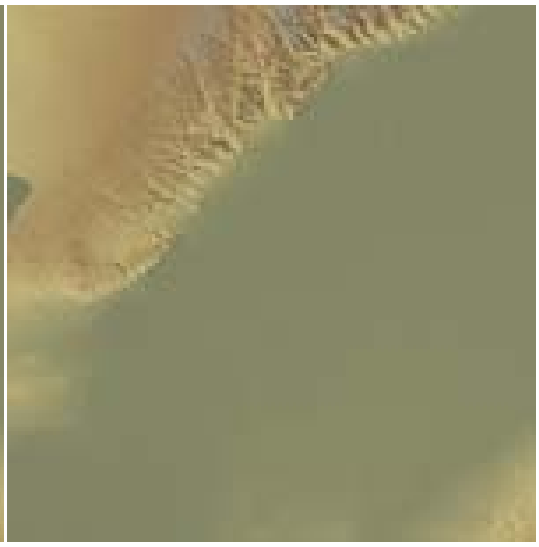


Figure 3.2 Void eliminated hgt data

Then we prepare for the Landsat ETM data with bands 7, 4, 2 in red, green and blue that can provide greatest distinction between city environment and countryside areas instead of natural colour composition, which is usually used primarily for display purpose and as background images.

Figure 3.3 Landsat bands composition by 7, 4&2

3.2.2 New Projection for Large Scale GIS

Map projections are attempts to portray the surface of the earth or a portion of the earth on a flat surface. The projection has some characteristics:

- Conformality
  - When the scale of a map at any point on the map is the same in any direction, the projection is conformal. Meridians (lines of longitude) and parallels (lines of latitude) intersect at right angles. Shape is preserved locally on conformal maps.
- Distance
  - A map is equidistant when it portrays distances from the centre of the projection to any other place on the map.

- Direction
  - A map preserves direction when azimuths (angles from a point on a line to another point) are portrayed correctly in all directions.
- Scale
  - Scale is the relationship between a distance portrayed on a map and the same distance on the Earth.
- Area
  - When a map portrays areas over the entire map so that all mapped areas have the same proportional relationship to the areas on the Earth that they represent, the map is an equal area map.

At the present, almost each GIS system is based on Gauss coordinate system. Gauss plane is a linear space, however the earthly space is nonlinear, so it was applied as spatial reference by traditional map projection.

Unlike the Earth, most other bodies in our system are defined with a positive-west longitude system. Although this seems like a trivial problem, most GIS and remote sensing applications cannot handle this longitude system. While a simple shift in longitude values fixes the problems, there is a lot of potential for confusion. We use a positive-east longitude system for our digital GIS files, but for output we use the positive-west convention for referencing and labelling.

As mentioned before, SRTM and Landsat data are accurate for terrain rending. However these two images belong to different projection system: Universal Transverse Mercator (UTM) and latitude longitude (LL). UTM grids are created by laying a square grid on the earth. This means that different maps will have different grids depending on the datum used (model of the shape of the earth). Latitude and longitude grids, on the other hand, are laid out with explicitly

physical frame of reference. Latitude is determined by the Earth's polar axis. Longitude is determined by the Earth's rotation. But there is no way to confirm the UTM coordinates except by calculation. Figure 3.4 and Figure 3.5 present UTM system, latitude and longitude system images separately.

Figure 3.4 UTM system

Figure 3.5 latitude and longitude system

After converting UTM to Lat/Lon system, we need to apply a new map projection extend the transition from earthly surface to horizontal surface. This is based on mother earth coordinates system and adopt isometric skive column. We make this by letting the map projection from factual ellipsoid reference space (B, L, H) to two-dimensional curving surface (B, L) along with the normal line. The new model establishes a practicably model under (B, L) metric space. That is described as following:

$$X=K \cdot L \qquad Y= K \cdot B \tag{3.1}$$

Where the unit of *B* and *L* is radian, *K* is the constant variable related with the scale of map which describes the length of one degree arc if meter is the unit of *X* and *Y*.

3.2.3 Real World Surface Generation Technologies

A commonly used format of digitally representing a real world surface is the Digital Elevation Model (DEM). Strictly speaking, a DEM specifically refers to a raster or regular grid of spot heights [USGS2003]. DEM consists of a sampled array of elevations for a number of ground positions at regularly spaced intervals to describe an axis-aligned grid of terrain. The distances between the sampled positions (i.e. the grid spacing) defines the resolution of a DEM and as well as defining the major determinant of its accuracy [USGS2003].

The DEM format is a compact and efficient way of representing terrain surfaces and lends itself well to computer computations. A variety of DEM are readily available from several government and private organizations. For instance, the United States Geological Survey (USGS) freely provides DEM data for the entire United States [USGS2003a]. The best resolution commonly available in the USGS DEM is 10m, with a vertical resolution of 1m. In addition, several private organizations (e.g. quarry owners) regularly update and maintain detailed (i.e. with fine resolution) DEM of their property for use in planning, quantification, and record-keeping [Zalubowski 2003].

A DEM is generally archived in one of two common digital formats. The standard method of archiving a DEM is through the use of numerical height maps. A numerical height map is simply a three dimensional array of numerical values. Each numerical value in the array indicates the elevation of the terrain's surface at that array position.

Another common method of archiving DEM is through the use of digital, greyscale image height maps. In greyscale images that represent DEM, the brightness of each image pixel corresponds to the height of the terrain at that point. Generally, dark regions on the image represent lower elevation values and lighter regions indicate high elevation values. The calibration of the greyscale (i.e. the numerical elevation values of pure black and pure white regions) is often implementation dependent.

3.3 Space Partition for Data Storage and Processing

3.3.1 Octree for Region Partition

An octree [Ahuja and Nash 1984][Chen and Huang 1988] is a way of subdividing 3D space, otherwise known as space partitioning. It allows you to only draw the part of your world/level/scene that is in your frustum (camera's view).

If you have a huge world with tens of thousands of triangles, it is not ideal to pass those all into OpenGL to be rendered. You only want to pass in the triangles that you can see (the camera's view). To do this, you want to subdivide your world into cubes that holds the triangle data for that area/region of the world. Then, instead of checking if every triangle is in your view, you can just check if a cube intersects your frustum, meaning it's in the view of your camera. This is one of the fastest ways to do this type of space partitioning.

Initially, the octree starts with a root node that has an axis aligned cube surrounding the entire world (Figure 3.6).

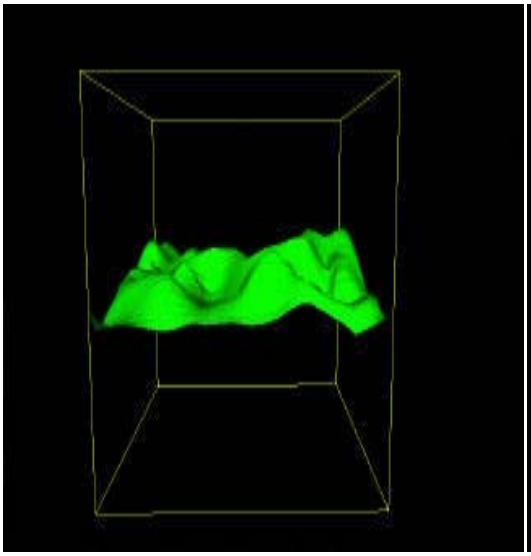


Figure 3.6 no partition

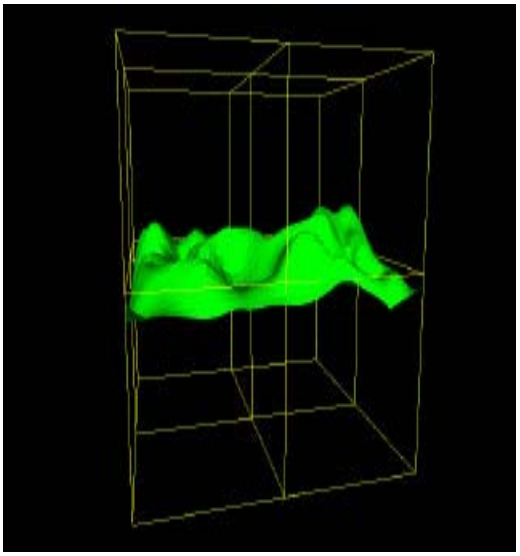


Figure 3.7 the traditional octree

So this root node now stores all the vertices in the world. Then we subdivide it into eight parts. Once we do this subdivision, there should be 8 cubes inside of the original root node's cube. That means 4 cubes on top and 4 on the bottom.

This is the first subdivision, but if we have 2, 3 or 4 subdivisions a good effective performance can be achieved.

3.3.2 Quadtrees for Region Partition

An important data structure which is used in spliting and merging algorithms is the quadtree. Figure 3.8 shows a quadtree and its relation to the image. Note that in graphics the quadtree is used in a region splitting algorithm which breaks a graphical image down recursively from the root node, which represents the whole image, to the leaf nodes which each represents a coherent region, which can be rendered without further hidden line elimination calculations. The same use is made of quadtrees for vision.Quadtrees impose one type of regular decomposition onto an image. To complete the segmentation process this must be followed by a merging phase. Thus the problem of finding adjacent neighbours to a given node

has been studied (Figure 3.9). The problem of one of the tree search and efficient algorithms have been published.

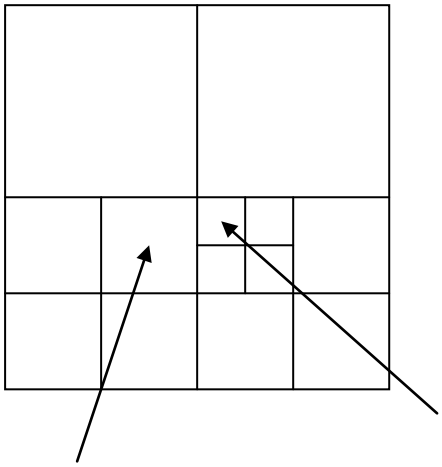


Figure 3.8 a quadtree

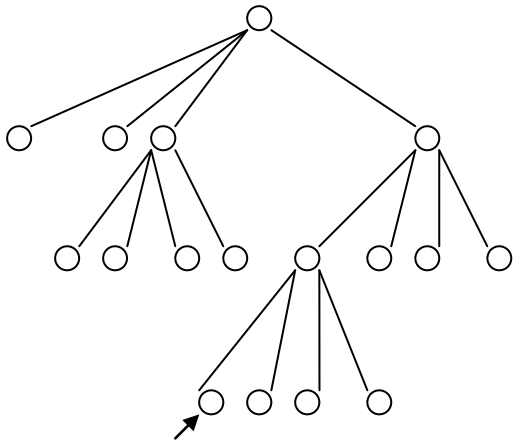


Figure 3.9 tree search algorithm

A characteristic of the split and merge method was the algorithm due to Horwitz and Plavidis [Horwitz and Pavlidis 1976]. This was based on the use of a segmentation tree, which is normally a quadtree. Each node, say k, in the tree corresponds to a square region of the image, and has stored with it a maximum M_k and minimum m_k brightness (or other search property) value. The algorithm starts with a cut set of nodes. This consists of five arrays storing the node properties: $\{x_k,y_k,size,M_k,m_k\}$. Note that only the cut set is stored, not the whole tree. For an image of resolution N x N pixels it is theoretically possible that each array requires N^2 entries, though, in practical cases, where regions do exist, and the requirement is much smaller. Merging is done by comparing adjacent groups of four nodes with a common parent. If they obey the criterion they are replaced by their parent node. Similarly splitting is done by breaking non uniform nodes into the four children nodes at the lower level (Figure 3.10). The process is constrained

by the tree structure, so, when all the splitting and merging is completed, the final cut set must be grouped into regions. This is essentially a further merging process.

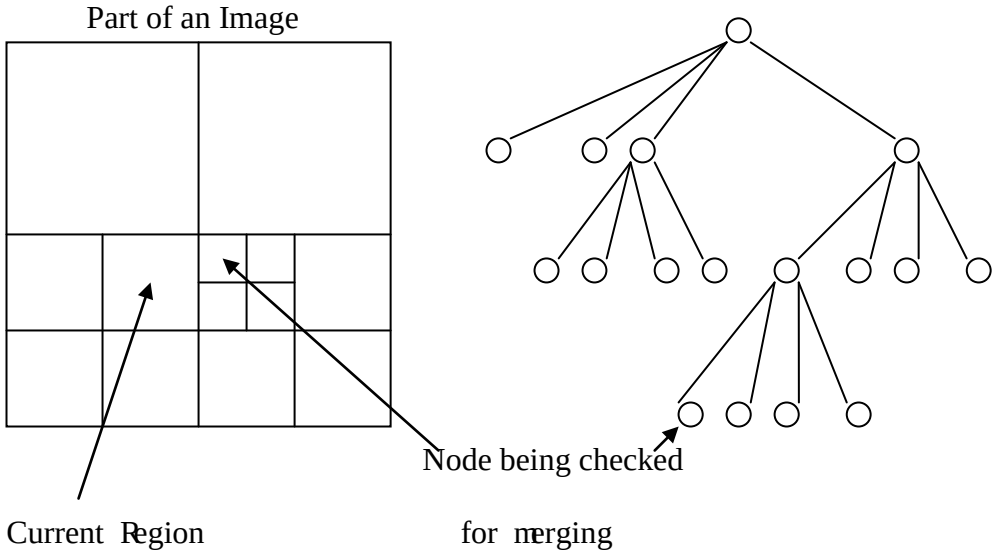


Figure 3.10 non uniform nodes to four children nodes

3.3.3 Constructing the Real World Geometry

The geometry of uneven, three dimensional surfaces is generally represented on the computer as a continuous mesh of polygons (usually triangles). The process of converting DEM data into a 3D terrain database thus requires interpretation of the input data and construction of a continuous, 3D polygon mesh corresponding to that data. The coordinates of each vertex in the constructed mesh can be directly obtained from the DEM if it is archived as a numerical height map. In cases where the DEM data is archived as a greyscale image, the extraction of coordinates from the height map requires two, implementation dependent calibration factors. The first is the numerical value of the horizontal spacing (i.e. distance) between sampled positions. This value defines the horizontal resolution of the grid by indicating how the distance between adjacent pixels in the input image must be interpreted. The second calibration factor

required to extract 3D coordinates from an image height map is the vertical scaling factor. This factor defines the vertical resolution of the image height map.

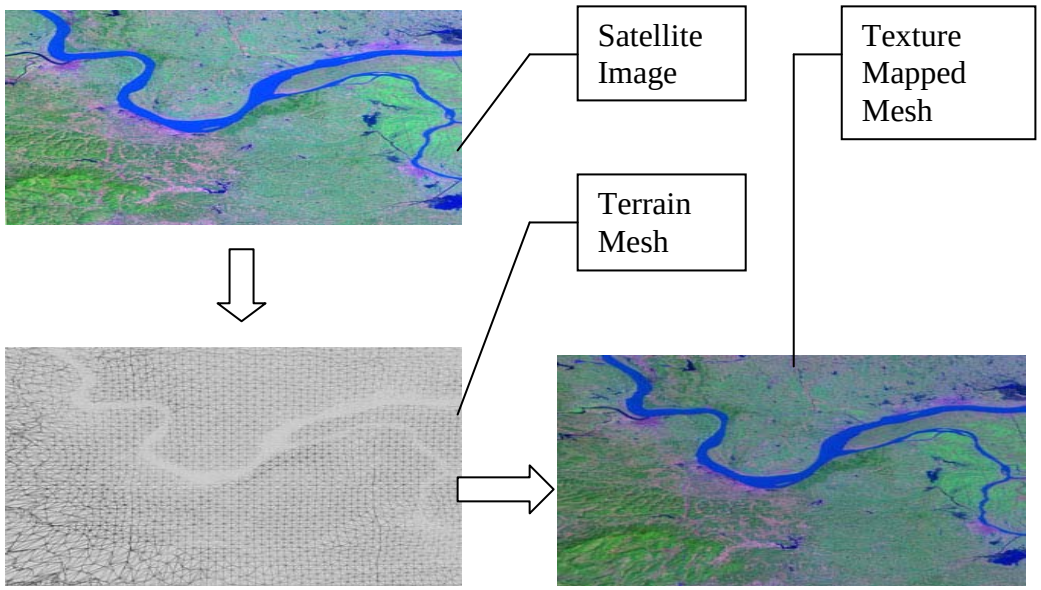


Figure 3.11 Texture mapping real world meshes

For instance, if the colours scale of a particular implementation assigns 0.0 to pure black and 1.0 to pure white and the vertical scaling factor is 1000m, then a pure white pixel on the greyscale image height map is interpreted to have an elevation of 1000m. A pure black pixel is similarly interpreted as a 0m elevation value. Shades of grey (colour values ranging between 0.0 and 1.0) are linearly interpolated to decipher the elevations indicated by each pixel in the image height map. Once the coordinates of the vertices are extracted from the image height map, a 3D polygon mesh can be constructed in a similar manner to that of numerical height maps. A greyscale image height map must thus be converted to a numerical height map before 3D mesh vertex coordinates can be extracted from the data. In both cases, the extraction of vertex coordinates from the height maps and construction of a corresponding 3D mesh generate a computer interpretable geometrical shape of the terrain segment that is archived in the input DEM.

3.4 3D Real World Modelling System

3.4.1 Nona Tree Space Partitions (NTSP) for DTM Modelling

The approach used to construct the geometry of a local earth region in three-dimensional space is to create a continuous rendered mesh of polygons, the process of which requires converting digital GIS data into geometrically-real world and a reliable accurate interpretation of the data. In order to develop an efficient space partition for real-time terrain generation capable of dealing with very large GIS data, and motivated by the widely used Octree algorithm [Ahuja and Nash 1984][Chen and Huang 1988], we propose the Nona tree space partitions (NTSP) algorithm in the current work. NTSP is useful for planar navagation, however, octree is applied to spatial data refinement. The centre of nine local regions has been assigned as a root node, which stores all the vertices of that local region. This arrangement makes it much easier to access GIS data for constructing the world in real-time, and for navigation towards neighbouring regions. The root node is divided into nine parts represented by the corresponding nodes, the process of which is the first subdivision. Each node that is newly created, can then be subdivided further to form a new subdivision. The process will go on until a threshold is reached. A suitable data structure is created for the use of split and merge algorithm as a quadtree [Ulrich 2000]. This partition scheme is able to perform an out-of-core rendering and to deal with massive GIS terrain data. Since the root node location is assigned in real-time and the reconstruction is made dynamically, the entire world space can be navigated.

The scheme implemented well with the ROAM algorithm. The key part in ROAM is a binary triangles tree structure which has been show in Figure 3.12. Because the original terrain is square, we need to use two triangles which are

right-angled and left-angled ones on the top level of the tree. Then the first right-angled can be divided into two descendants. Depending on the refinement of terrain, this subdivision will continue for *n* levels. At the end, the number of triangles after n levels of division is 2(*n*+1)-1. Every level of subdivision does not replace the level above and only the bottom level is drawn while the rest can be utilized for frustum and collision detection.

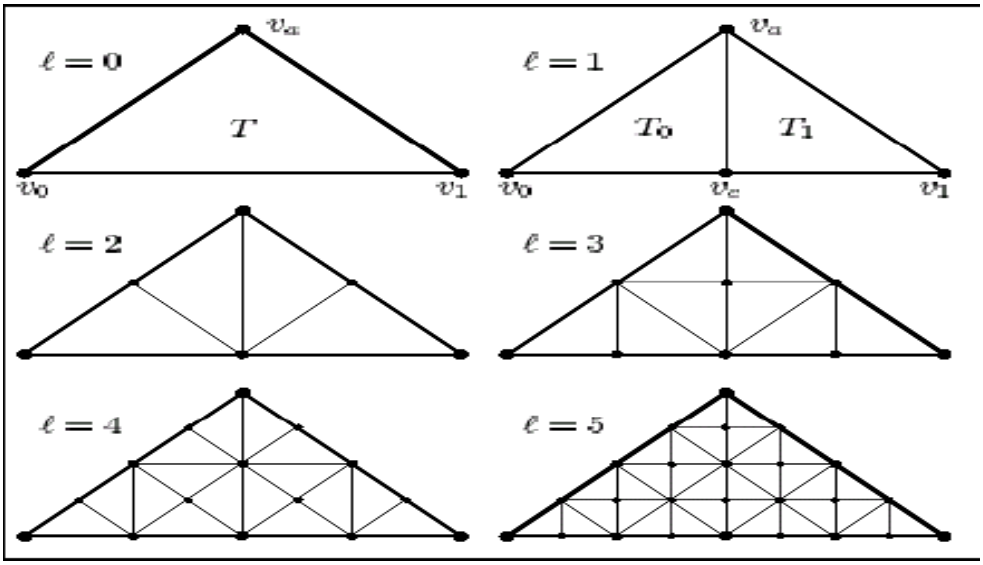


Figure 3.12 Levels 0-5 of a binary triangle tree

At the same time, each triangle is stored as a node. Every vertex can be donated as V_p (V_x, V_y, V_z), where V_z is the height at the domain coordinate of vertex v, which is defined by the value in elevation height map, V_z=Z (v). The depth of tree node to be future split is calculated by formulation 3.2 where, δ is the visual tolerance:

$$| Z_{\text{centre}} - (Z(V_{\text{left}}) + Z(V_{\text{right}})/2 | \leq \delta \quad (3.2)$$

Every node will be connected with its parent and children. It will be easy for us dealing with the spilt and merge operations which is show in Figure 3.13.

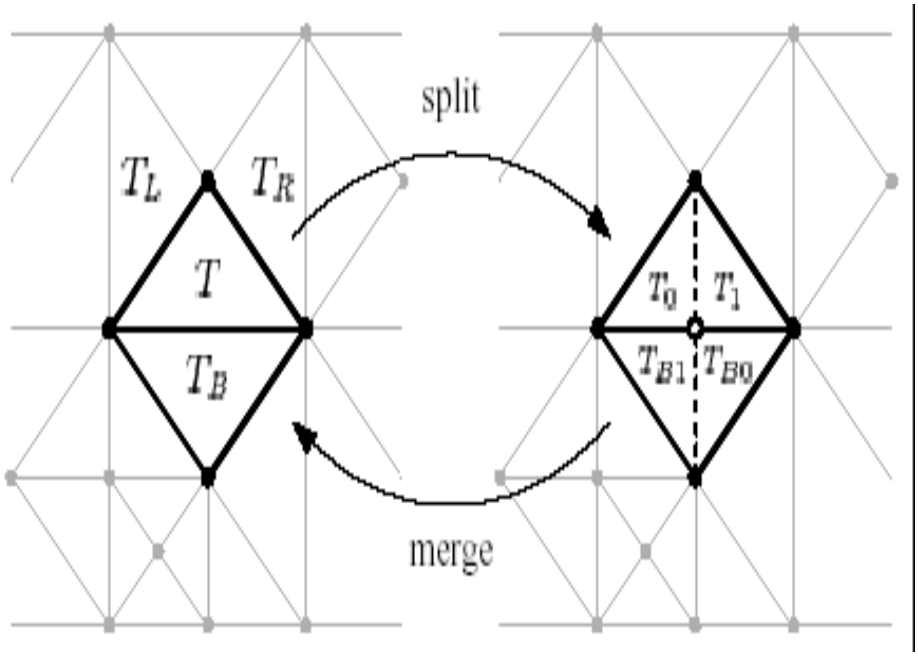


Figure 3.13 Split and merge operations on a triangle diamond

Figure 3.14 shows a typical nine-terrain-node structure used in the algorithm. The observer's perspective position is in N_5, which is the centre of the local region. As the observer crosses over any of the boundaries of N_5, a new node arrangement will be made; the updated root node is assigned to the node located in the central position of the region. Figure 3.15 shows a situation where the navigation conducted by an observer moves towards the right. The relative locations of N_1, N_4 and N_7 will be discarded and replaced by N_2, N_5 and N_8. The nodes N_1, N_4 and N_7 will reappear in the right side but with GIS data of the neighbouring region. As we only need to use the nine nodes for world visualisation for navigation, the method is very efficient and a whole world can be visualized as long as the necessary GIS data available.

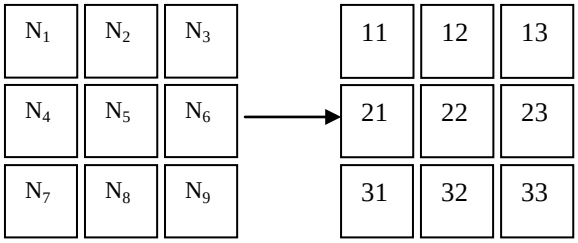


Figure 3.14 Basic nine nodes of a local region

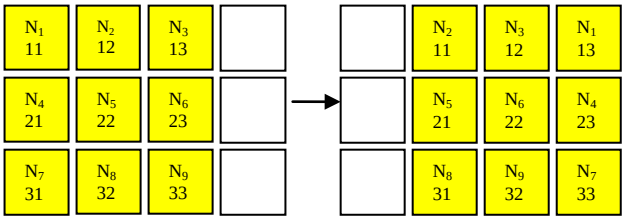


Figure 3.15 Moving to right

3.4.2 Pyramidal Data Arrangement Structure

As demonstrated above, the surface generated using the method therefore is capable of dealing with a very large earth region. When an observer moves closer to the Earth surface region for details, it is necessary to use a detailed or high resolution data set for the reconstruction. An in-core process with a pyramidal data arrangement structure is proposed for processing the terrain information and further for constructing 3D terrain geometry using NTSP algorithm, as shown in Figure 3.16. This work can be considered as an expansion of Liu and Miao's 2D projection map to 3D terrain reconstruction [Liu and Miao 2005]. The node N_5 is used for illustration. At layer one, we can display a large scale terrain reconstructed using a low resolution data set. A higher resolution model is created with higher layers; as indicated by Layer two with nine region elements, Layer three representing 81 region elements, and so on.

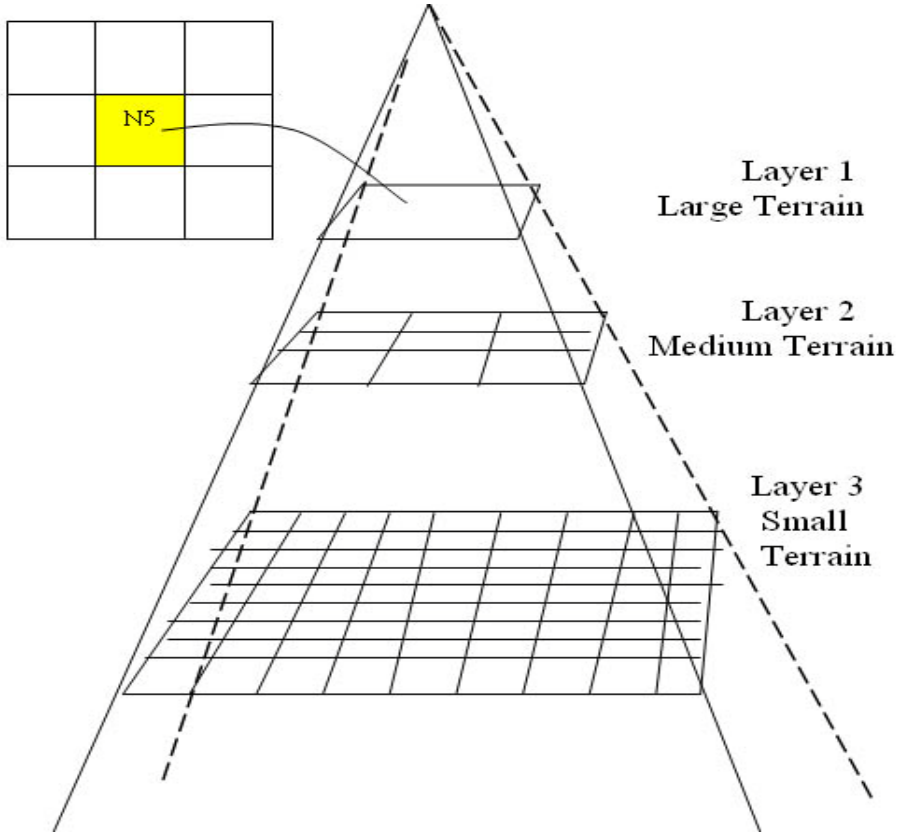


Figure 3.16 Pyramidal data arrangement structure

3.4.3 Terrain Data Interpolation

It is easy to understand that a high resolution urban environment needs more grids in comparison to a low resolution urban environment if both environments are in the same level of details. Modelling a detailed urban environment means more data has to be loaded during the initialization, which will affect the performance of the system. Another problem is that the files contain a large amount of data will be more difficult to be maintained or modified. In order to conquer these disadvantages, a hierarchical structure was adopted in our system. While creating a detailed urban environment, only part of the grid squares are initialized with a value. These squares are well-distributed in the 2D plane. Once the data is loaded into the memory, interpolation algorithm is called to fill the vacancies firstly according to the *x* direction, and then the *y* direction, or

vice verse. Different interpolation algorithm can be used here according to different requirements. Linear interpolation is the simplest and the fastest method while the methods using curve interpolations can be used to create a smoother surface. An illumination of the linear interpolation of the terrain data is shown in Figure 3.17.

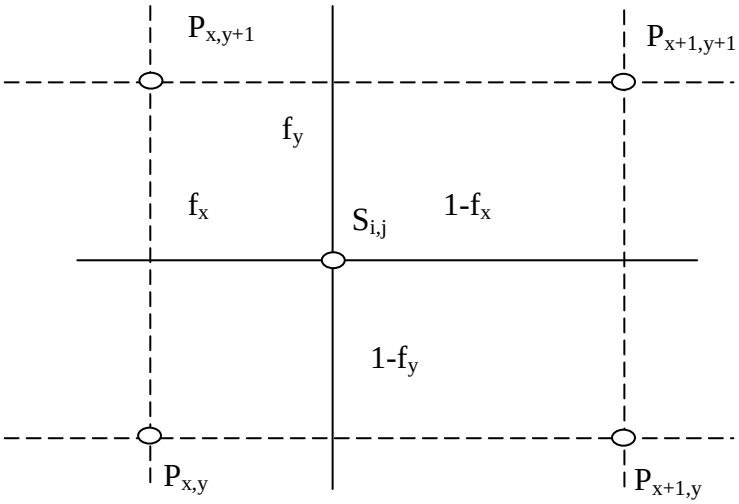


Figure 3.17 Linear interpolations of the terrain data.

In Figure 3.17, $P_{x,y}$, $P_{x+1,y}$, $P_{x,y+1}$ and $P_{x+1,y+1}$ are the grid squares that are assigned a value during the initialization. $S_{i,j}$ is the grid square that is not initialized with a value and is located in the area defined by the four *P*-grids. f_x and f_y are the distances from $S_{i,j}$ to the edges formed by ($P_{x,y}$, $P_{x,y+1}$) and ($P_{x,y+1}$, $P_{x+1,y+1}$). Choose linear interpolation as an example, the elevation value of $S_{i,j}$ can be found via the equation 3.3:

$$S_{i,j} = (P_{x,y+1} * f_x + P_{x+1,y+1} * (1 - f_x)) * f_y + (P_{x,y} * f_x + P_{x+1,y} * (1 - f_x)) * (1 - f_y) \qquad (3.3)$$

3.4.4 Map Projections Implementation

The next task for reconstructing the earth region is to implement map projections, which is to portray the surface of the earth or a portion of the earth on

a flat surface. At present time, almost all GIS systems are based on a Gaussian coordinate system, which is a Gauss plane, a linear space. However since the Earth space is nonlinear, the spatial reference and consideration must be addressed by the projection. Unlike the Earth, most other objects are defined with a positive-west longitude system [PW]. Although this seems like a trivial problem, most GIS and remote sensing applications cannot handle this longitude system. A simple shift in longitude values can solve the problems. In the current work, a positive-east longitude system is used and the positive-west convention for output referencing and labelling.

3.4.5 Experiment Results

The first experiment in our test has been chosen at the famous tourist island Skye in Scotland. At 1,656 km², Skye is the second largest island in Scotland after Lewis and Harris. The island is renowned for its spectacular scenery, vibrant culture and heritage.

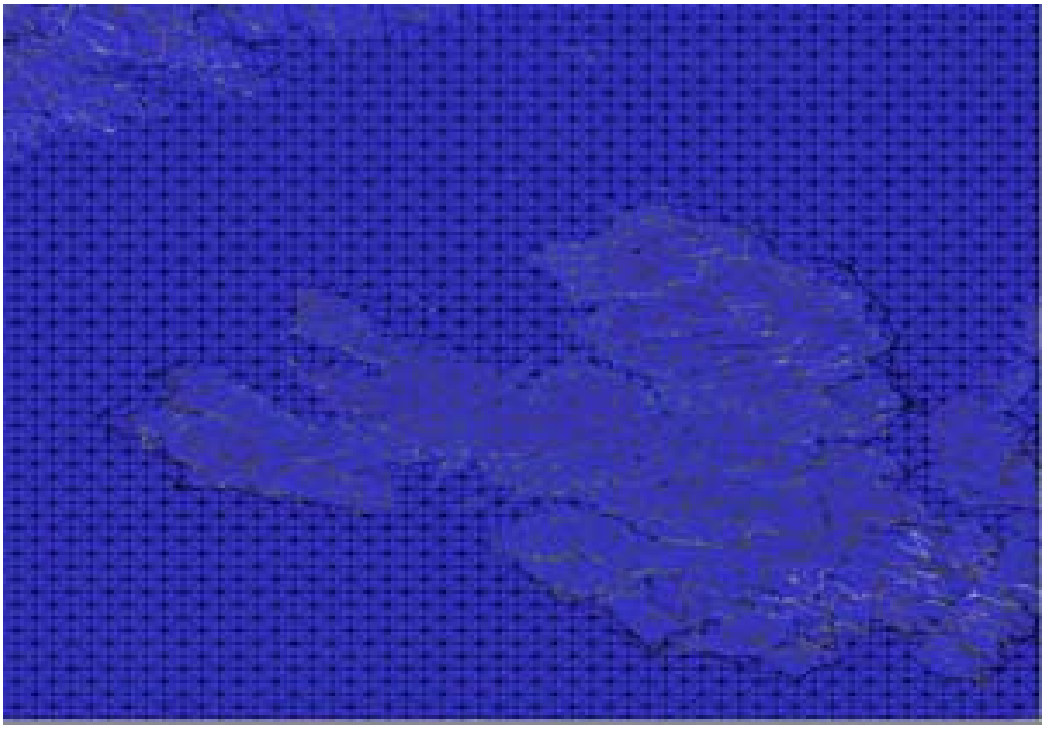


Figure 3.18 The Skye terrain geometry

Figure 3.18 shows the Skye terrain geometry which is generated by our digital terrain model. Figure 3.19 shows the far distance view of the Skye region in Scotland reconstructed by the method proposed in the current work. As shown in the Figure 3.19, the generated terrain geometry (Figure 3.18) is overlaid by satellite image corresponding to the same geographic area represented as the input to the virtual Earth model. This test shows that our method is suitable for constructing large scale areas.

Figure 3.19 The far distance view of the Skye region reconstructed

The second experiment test has been conducted for constructing the 3D real environment of Chepstow in Wales.

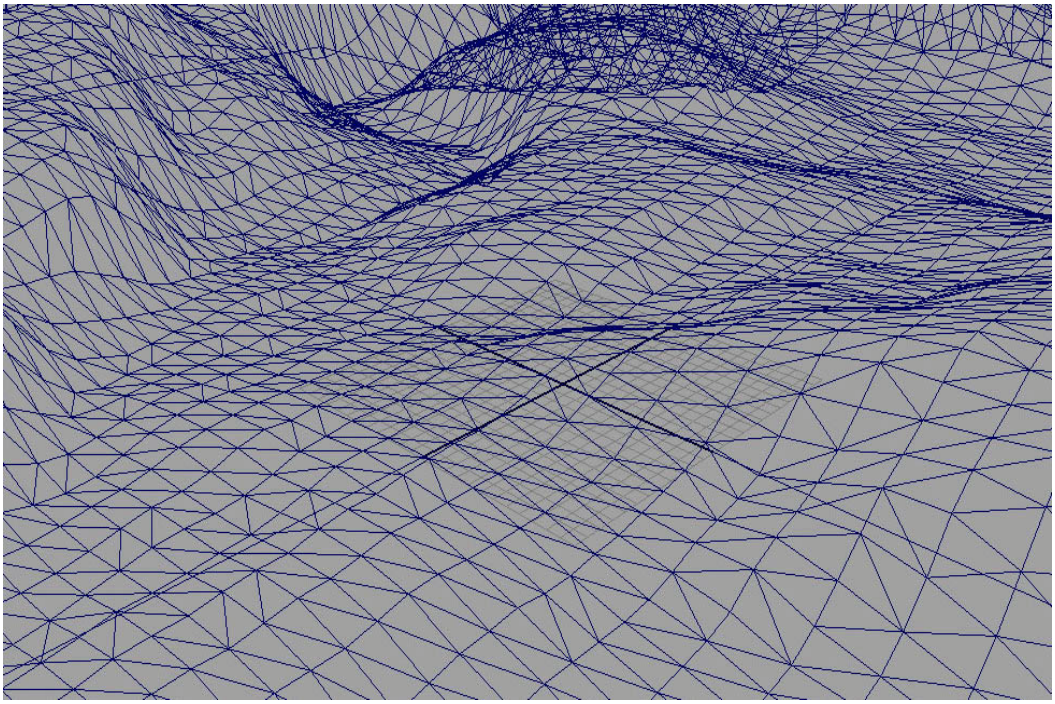


Figure 3.20 The Chepstow terrain geometry

Figure 3.21 shows the close distance view of the Chepstow region in Wales reconstructed by the method proposed in the current work. This test gives evidence that our approach is still available for modelling 3D urban environment.

Figure 3.21The close distance view of the Chepstow region reconstructed

Figure 3.22 Large scale terrain

Figure 3.23 Medium scale terrain

Figure 3.24 Small scale terrain

The last test has been conducted for generating a digital terrain model of a region of middle and lower reaches of Yangtze area in China. This result shows our proposed NTSP algorithm and Pyramidal data arrangement structure are capable of generating 3D real world.

3.5 Conclusion

In this chapter, we have presented a new dynamic terrain modelling with GIS data with NTSP algorithm and pyramidal data arrangement structure. The dynamic model is constructed using real-world measurement data of GIS, in terms of digital elevation data and satellite image data. NTSP is proposed for very large data assessing, terrain modelling and visualisation. Pyramidal data arrangement structure is used for dealing with requirement of terrain details with different resolution.

Chapter 4 Active Contour based Image Segmentation for Object Identification

Once a digital terrain has been generated, from the flood hazard warning application point of view, the following tasks need to be conducted: To identify which area needs to be reconstructed, adaptive image segmentation algorithm is necessary. In order to identify environmental objects, intelligent shape match method has been proposed which is useful for shape feature extraction and matching. On the other hand, recognise the river object in satellite image is an important step to build our flood hazard warning system. For this purpose, we propose a novel multi-resolution vector-valued framework for images segmentation based on active contours.

The proposed multi-resolution vector-valued framework has two advantages:

- It is able to initialize a number of snakes automatically in complicated satellite images based on colour and texture features.
- The combined low resolution segmentation and high resolution optimization algorithm allows for a rapid evolution of the contour and a convergence to its final configuration with a small number of iterations.

The proposed environmental objects identification method is efficient for finding primitives which contain similar parts and counting the number of similar portions automatically.

4.1 Introduction

Image analysis, usually, refers to a process of images provided by a computer in order to find the objects within the image. Image segmentation is one of the most critical tasks in automatic image analysis. It consists of subdividing an image into its constituent parts as well as extracting them.

A great variety of segmentation algorithms have been developed in the last few decades; but more of these algorithms cannot be really applied to all images. Some of them are not suitable for some particular situations, especially in satellite images which, often, contain different textured regions or varying background, and are often subjected to illumination changes or environmental effects.

Active contours models have become effective tools for the extraction of region of interests (ROI), which were widely investigated for overcoming the limitations of traditional methods. The existing active contour models can be categorized into two classes: edge-based models [Caselles et al 1997][Li et al 2005][Malladi et al 1995] and region-based models [Chan and Vese 2001][Paragios and Deriche 2002][Tsai et al 2001]. These two types of models both have their pros and cons, and the choice of them in applications depends on different characteristics of images.

In this chapter, we have proposed a novel multi-resolution vector-valued framework for robust satellite images segmentation and a novel environmental objects identification which has following advantages:

- We define the vector-valued scanning algorithm which can automatically assign initial snakes as if their specific values inside the calculated vector of image features.
- An active contours model is used to detect object boundaries. This

model can be easily incorporated into various prior knowledge such as shape and region distribution. The final contours are quite regular, which are convenient for further applications, such as shape analysis, classification, and recognition.

- The framework can increase high resolution image processing speed by detecting object boundary in low resolution. Then the boundary result from the low resolution is used to initialize the front for the high resolution. Meanwhile, a fast scheme passing the contour from low resolution to high resolution is developed. Furthermore, an efficient optimization method at the finer level is proposed, which propagates the curve towards the real boundary. The optimization approach is rapid because the initial curve is close to desirable result.
- Environmental objects can be identified rapidly by intelligent shape match method which is useful for finding primitives which contain similar parts and counting the number of similar portions automatically.

This chapter is organized as follows: Section 1 gives an introduction about active contour based image segmentation for object identification; Section 2 introduces several image processing techniques; Section 3, Section 4 and Section5 review threshold segmentation for object recognition, identification by optimal edge detectors and region segmentation for extracting objects; Section 6 describes the image fusion method for integrating multiple images into a composite image; Section 7 explains our proposed multi-resolution vector-valued framework; Section 8 discusses a novel environmental objects identification by using intelligent shape match method; Section 9 draws the conclusion.

4.2 Satellite Images Processing Techniques

In today's world of advanced technology where most satellite images are recorded in digital format virtually, all image interpretation and analysis involve some elements of digital processing. Digital image processing may involve numerous procedures including formatting and correcting of the data, digital enhancement to facilitate better visual interpretation, or even automated classification of targets and features entirely done by computer [Campbell 1997].

The development and application of various remote sensing platforms result in the production of huge amounts of satellite image data. Therefore, there is an increasing need for effective querying and browsing in these image databases. In order to take advantage and make good use of satellite images data, we must be able to extract meaningful information from the imagery. Indeed, interpretation and analysis of satellite imagery involves the identification and/or measurement of various targets in an image in order to extract useful information about them [Lillesand and Kiefer 1994]. Targets in satellite images may be any feature or object which can be observed in an image, and have the following characteristics:

• Targets may be a point, line, or area feature. This means that they can have any form, a bus in a parking lot or plane on a runway, a bridge or roadway, a large expanse of water or a field.

• The target must be distinguishable; it must contrast with other features around it in the image.

The satellite image processing techniques available in image analysis systems can be classified into the following categories:

• Pre-processing

• Image Transformation

• Image segmentation and Analysis

The pre-processing techniques involve different operations applicable to the satellite images among which one quotes: analysis and extraction of information; data correction for sensor irregularities and unwanted sensor or atmospheric noise; data conversion. So these techniques operations accurately represent the reflected or emitted radiation measured by the sensor; and geometric distortions correction due to sensor-Earth geometry variations.

Image transformation involves changing nonlinear Universal Transverse Mercator (UTM) based image coordinate into linear latitude longitude (LL) based image coordinate.

Image segmentation and analysis operations are used to digitally identify and classify pixels, blocks or windows in the data. Segmentation is usually performed on multi-channel data sets and this process assigns each pixel in an image to a particular class or theme based on different characteristics of the pixel brightness values.

There are a variety of approaches taken to perform digital segmentation. We will describe two generic approaches, normally the must used in literature: edge and region segmentation.

4.3 Threshold Segmentation for Object Recognition

This method consists in comparing the measure associated to each pixel to one or some threshold in order to determine the class which the pixel belongs to. The attribute is generally the grey level, although colour or a simple texture descriptor can also be used. Threshold may be applied globally across the image (static threshold) or may be applied locally so that the threshold varies dynamically across the image.

Several algorithms have been proposed for threshold segmentation of satellite images, proposing thus an automatically selection of an adequate threshold for different satellite image class.

In these algorithms developed below, one uses the following partial sums:

For j=0,……, n;

$$\begin{cases} A_j = \sum_{i=0}^{j} h(i) & B_j = \sum_{i=0}^{i} ih(i) \\ C_j = \sum_{i=0}^{j} i^2 h(i) & D_j = \sum_{i=0}^{i} i^3 h(i) \end{cases} \quad (4.1)$$

where h(i) is the image histogram.

Minimum algorithm is developed by Prewitt and Mendelsohn [Prewitt and Mendelsohn 1996]. It consists of choosing t (threshold) in order to minimize h(t) between two maxima of the histogram.

Inter-mode algorithm is an alternative of the minimum algorithm [Prewitt and Mendelsohn 1996], it consists in choosing the threshold t as the average of the two grey levels corresponding to the two histogram maximum h (max_1) and h (max_2), relating to the object and its background, that is to say: t= (max_1+max_2) /2.

Moment algorithm designed by Fan et al [Fan et al 2001], chooses t so that the binary image and its original have the same three first moments. Thus, the value of t is that which brings best the fraction A_t/A_n closer to the value of x_0

$$\text{where: } \begin{cases} x_0 = \dfrac{\frac{B_n}{A_n}+\frac{x_2}{2}}{2\sqrt{\left(\frac{x_2^2}{4}-x_1\right)}} \\ x_1 = \dfrac{B_n D_n - C_n^2}{A_n C_n - B_n^2} \\ x_2 = \dfrac{B_n C_n - A_n D_n}{A_n C_n - B_n^2} \end{cases} \quad (4.2)$$

Threshold entropy established by Kapur and Al [Kapur et al 1985], consists

of evaluating, for all j= 0 ..... n, the following expression:

$$\frac{E_j}{A_j} - \log(A_j) + \frac{(E_n - E_j)}{(A_n - A_j)} - \log(A_n - A_j) \tag{4.3}$$

The value of the threshold t is that of j for which this expression is minimal.

The inter-means algorithm developed by Otsu [Otsu 1978] maximizes the expression:

For j = 0 … n-1

$$X_j = A_j(A_n - A_j)(\mu_j - \nu_j)^2 \tag{4.4}$$

The threshold value of t is that of j which maximizes X_j. This mathematical formulation has for effect to position the threshold t in the middle of the averages of the two image classes.

4.4 Identification by Optimal Edge Detectors

Edges detectors for object identification are a particularly simple and effective means for increasing geometric detail in an image. It is performed by first detecting edges and then either adding these back into the original image to increase contrast in the vicinity of an edge, or highlighting edges using saturated (black, white or colour) overlays on borders. Indeed, edges in images are areas with strong intensity contrasts – a jump in intensity from one pixel to the next.

Edge detecting an image significantly reduces the amount of data and filters out of useless information, while preserving the important structural properties in an image. The fundamental importance of line and edge information in images has been recognized [Cumani 1991].

Indeed, local features, such as lines and edges, can describe the structure of a scene relatively independently on the illumination.

Image segmentation techniques based on edge have long been in use.

Although a variety of methods of edge detection have been suggested, and something still may be grouped into two categories: derivative methods [Davies 1990] and optimal filtering method [Demigny and Kamle 1997].

4.4.1 Derivative Methods

In computer vision, edge detection is traditionally implemented by convolving the signal with some form of linear filter, usually a filter that approximates a first or second derivative operator. An odd symmetric filter will approximate a first derivative, and peaks in the convolution output will correspond to edges (luminance discontinuities) in the image. An even symmetric filter will approximate a second derivative operator. Zero-crossings in the output of convolution with an even symmetric filter will correspond to edges; maxima in the output of this operator will correspond to tangent discontinuities, often referred to as bars, or lines.

4.4.1.1 First Order Derivative Methods

Most edge detection methods work on the assumption that an edge occurs where there is a discontinuity in the intensity function or a very steep intensity gradient in the image. Using this assumption, if we take the derivative of the intensity values across the image and find points where the derivative is a maximum, we will have marked our edges.

We will show thereafter three types of operators which belong to first order derivative methods.

- Sobel operator:

The Sobel operator performs a 2D spatial gradient measurement on an image and so emphasizes regions of high spatial frequency that correspond to edges. Typically it is used to find the approximate absolute gradient magnitude at each

point in an input greyscale image.

In theory at least, the operator consists of a pair of 3×3 convolution kernels. One kernel is simply the other rotated by 90°.

These kernels are designed to respond maximally to edges running vertically and horizontally relative to the pixel grid, one kernel for each of the two perpendicular orientations. The kernels can be applied separately to the input image, to produce separate measurements of the gradient component in each orientation.

-Roberts operator:

The Roberts Cross operator performs a simple, quick to compute, 2D spatial gradient measurement on an image. It thus highlights regions of high spatial frequency which often correspond to edges. In its most common usage, the input to the operator is a greyscale image, as is the output. Pixel values at each point in the output represent the estimated absolute magnitude of the spatial gradient of the input image at that point.

In theory, the operator consists of a pair of 2×2 convolution kernels. One kernel is simply the other rotated by 90°. This is very similar to the Sobel operator.

These kernels are designed to respond maximally to edges running at 45° to the pixel grid, one kernel for each of the two perpendicular orientations. The kernels can be applied separately to the input image, to produce separate measurements of the gradient component in each orientation.

-Prewitt operator:

The Prewitt operator, similarly to the Sobel, approximates the first derivative.

4.4.1.2 Second Order Derivative Methods

All of the previous edge detectors have approximated the first order derivatives of pixel values in an image.

It is also possible to use second order derivatives to detect edges. Indeed, the second order derivative methods, optimal edges (maxima of gradient magnitude) are found by searching for places where the second derivative is zero. The isotropic generalization of the second derivative to two dimensions is the Laplacien.

-Laplacien operator:

The Laplacian is a 2D isotropic measure of the 2nd spatial derivative of an image. The Laplacian of an image highlights regions of rapid intensity change and is therefore often used for edge detection

The Laplacian of a function f(x,y), denoted by $\sigma^2 f(x,y)$, is defined by :

$$\sigma^2 f(x,y) = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2} \tag{4.5}$$

We can use discrete difference approximations to estimate the derivatives and represent the Laplacian operator with the convolution mask 3 x 3 for 4-neighbourhoods and 8-neighbourhoods.

Using one of these kernels, the Laplacian can be calculated using standard convolution methods.

Because these kernels are approximating a second derivative measurement on the image, they are very sensitive to noise. To counter this, the image is often Gaussian smoothed before applying the Laplacian filter. This pre-processing step reduces the high frequency noise components prior to the differentiation step.

-Laplacian of Gaussian operator (LoG):

In fact, since the convolution operation is associative, we can convolve the Gaussian smoothing filter with the Laplacian filter first of all, and then convolve this hybrid filter with the image to achieve the required result. By doing the precess mentioned, this way has two advantages:

-Since both the Gaussian and the Laplacian kernels are generally much smaller than the image, this method usually requires far fewer arithmetic operations.

-The LoG (Laplacian of Gaussian) kernel can be recalculated in advance so only one convolution needs to be performed at run-time on the image.

The Gaussian distribution function in two variables, g(x,y), is defined by :

$$g(x,y) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \tag{4.6}$$

where σ is the standard deviation representing the width of the Gaussian distribution.

The LoG operator calculates the second spatial derivative of an image. This means that in areas where the image has a constant intensity (i.e. where the intensity gradient is zero), the LoG response will be zero.

In the vicinity of a change in intensity, however, the LoG response will be positive on the darker side, and negative on the lighter side. This means that at a reasonably sharp edge between two regions of uniform but different intensities, the LoG response will be:

-zero at a long distance from the edge,

-positive just to one side of the edge,

-negative just to the other side of the edge, -zero at some point in between, on the edge itself.

It is possible to approximate the LoG filter with a filter that is just the difference of two differently sized Gaussians. Such a filter is known as a DoG filter (Difference of Gaussians).

4.4.2 Optimal Filtering Method

These methods consist in the following steps: given a model of an ideal edge and the type of the detection's operator, we seek the optimal filter which obeys the criteria and allows the detection of this model in a given context. The approach used as reference for a great number of methods remains the canny approach [Demigny and Kamle 1997]. It derives an optimal operator according to three criteria.

Canny detector:

Among the whole edge detection methods, the Canny edge detector is the most rigorously defined operator and is widely used. The popularity of the Canny edge detector can be attributed to its optimality according to the three criteria of good detection, good localization, and single response to an edge. In fact, in 1983, Canny proposed a theoretical study of edge detection. It is the first to formalize the three criteria, which must validate an edge detector.

The current standard edge detection scheme widely used around the world is the Canny edge detector. Canny treated edge detection as a signal processing problem and aimed to design the optimal edge detector. He formally specified an objective function to be optimized and used this to design the operator.

The objective function was designed to achieve the following optimization constraints:

- Maximize the signal to noise ratio to give good detection. This favors the marking of true positives.

- Achieve good localization to accurately mark edges.

- Minimize the number of responses to a single edge. This favors the identification of true negatives, that is, non-edges are not marked.

The overall edge detection procedure developed by Canny was as follows:

1. Find the maxima of the partial derivative of the image function I in the direction orthogonal to the edge direction, and to smooth the signal along the edge direction. Thus Canny's operator looks for the maxima of where However, many implementations of the Canny edge detector actually approximate this process by first convolving the image with a Gaussian to smooth the signal, and then looking for maxima in the first partial derivatives of the resulting signal (using masks similar to the Sobel masks). Thus we can convolve the image with 4 masks, looking for horizontal, vertical and diagonal edges. The direction producing the largest result at each pixel point is marked. Record the convolution result and the direction of the edge at each pixel.

2. Perform non-maximal suppression. Any gradient value that is not a local peak is set to zero. The edge direction is used in this process.

3. Find connected sets of edge points and form into lists.

4. Threshold these edges to eliminate insignificant edges. Canny introduced the idea of threshold hysteretic. This involves having two different threshold values, usually the higher threshold being 3 times the lower. Any pixel in an edge list that has a gradient greater than the higher threshold value is classed as a valid edge point. Any pixels connected to these valid edge points that have a gradient value above the lower threshold value are also classed as edge points. That is, once you have started an edge you don't stop until the gradient on the edge has dropped considerably. The purpose of the criteria defined by canny is to find an

analytical expression for the optimal filter for edge detection. However, the experiment shows that this filter presents two major drawbacks: the first one concerns the implementation of this filter, which is difficult. The second one relates to the errors of discretization and quantification of the filter, which involves a distortion of the obtained edges.

4.5 Region Segmentation for Extracting Objects

Another way of extracting and representing information from an image is to group pixels together into regions of similarity. One would group pixels together according to the rate of change of their intensity over a region. In fact, the regions formation can be realized in two different ways: (i) split and merge process founded on some global criteria, (ii) a points aggregation procedure based on some local similarity criteria. Due to the global level of the analysis, in a split and merge process the local information is not considered. On the other hand, a point's aggregation to a region needs to take into account some global information of the region, as well as some local information relative to the pixel. For this reason, the region extraction through point's aggregation approach has been retained.

4.5.1 Region Growing Segmentation

A range of image segmentation algorithms are based on region growing [Adams and Bischof 1994][Hijjatoleslami and Kittler 1998]. Region growing algorithms take one or more pixels, called seeds, and grow the regions around them based upon a certain homogeneity criteria. If the adjoining pixels are similar to the seed, they are merged with them within a single region. The process continues until all the pixels in the image are assigned to one or more regions.

Indeed, region growing is one of the most region based segmentation algorithms. It starts by choosing a starting point or seed pixel. The most habitual way is to select these seeds by randomly choosing a set of pixels in the image, or by following a priori set direction of scan of the image.

The region grows by successively adding neighboring pixels that are similar, according to a certain homogeneity criterion, increasing step by step the size of the region. The growing process is continued until a pixel and the average of the region is small. The growing process is continued until a pixel not sufficiently similar to be aggregated is found. It means that the pixel belongs to another object and the growing in this direction is finished. When there is not any neighboring pixel which is similar to the region, the segmentation of the region is completed.

4.5.2 Split and Merge Segmentation

One of the basic properties of segmentation is the existence of a predicate P which measures the region homogeneity. If this predicate is not satisfied for some region, it means that region is inhomogeneous and should be split into sub-regions. On the other hand, if the predicate is satisfied for the union of two adjacent regions, then these regions are collectively homogeneous and should be merged into a single region.

A way of working toward the satisfaction of these homogeneity criteria is the split and merge algorithm [Chang and Li 1994] [Adams and Bischof 1994]. This technique consists, as their name denotes, of two basic steps. First, the image is recursively split until all the regions verify a homogeneity criterion. Next, in a second step, all adjacent regions are reassembled of way that resulting regions satisfy the homogeneity criterion.

A quad-tree structure is often used to affect the step of splitting: it is based

on the recursive decomposition of the regions that does not verify the homogeneity criterion into four squared sub-regions, starting from the whole image. Therefore, an inverse pyramidal structure is built. The merging step consists on merging the adjacent blocks which represent homogenous regions but have been divided by the regular decomposition.

4.6 Image Fusion

Extraordinary advances in sensor technology and communication have brought a need for processing techniques that can effectively combine information from different sources into a single composite for interpretation.

Image fusion provides the means to integrate multiple images into a composite image that is more suitable for the purposes of human visual perception and computer processing tasks such as segmentation, feature extraction and target recognition [Abidi and Gonzalez 1992].

4.6.1 Satellite Image Fusion

In the early days of analogical remote sensing when the only remote sensing data source was aerial photography, the capability for data fusion from different sources was limited. Today, with most data available in digital format, from a wide array of sensors, data fusion is a common method used for interpretation and analysis. Data fusion fundamentally involves the combining or merging of data from multiple sources in an effort to extract better and/or more information. This may include data that are multi-temporal, multi-resolution, multi-sensor, or multi-data type in nature.

Imagery collected at different times is integrated to identify areas of change. Multi-resolution data merging is useful for a variety of applications. The merging

of data of a higher spatial resolution with data of lower resolution can significantly sharpen the spatial detail in an image and enhance the discrimination of features.

Data from different sensors may also be merged, bringing in the concept of multi-sensor data fusion. An excellent example of this technique is the combination of multispectral optical data with radar imagery. These two diverse spectral representations of the surface can provide complementary information. The optical data provide detailed spectral information useful for discriminating between surface cover types, while the radar imagery highlights the structural detail in the image.

4.6.2 Fusion of Edge and Region Approaches

The application of the space segmentation based on the edge detection provides precise and open edges, whereas methods based on the growing region techniques offer related areas but with less precise defined edges. It is thus interesting to be able to combine these approaches in order to obtain closed areas with quite localized edges.

There are various techniques for image fusion, even at the pixel level. The fusion technique proposed in this paper is a cooperative fusion [Pohl and Genderen 1998]. The principle of this technique is described below:

Let us note C the whole of the pixels obtained by the edge segmentation approach and Cr the whole of the pixels resulting from region segmentation approach. The approach will allow correcting the edges Cr using the edges C (here they have used 8-neighbourhood Laplacian). On the other hand, this approach makes it possible to close C with information provided by Cr.

In fact, the fusion principle consists in a growth of regions that take in

account edges that has been extracted beforehand in the picture original. This can be considered like a constraint according to the more important role that takes edges.

4.7 Multi-resolution Vector-valued Segmentation Framework

We have proposed a novel multi-resolution vector-valued framework for satellite images segmentation based on active contours. There are four core components of this framework: Vector-valued scanning algorithm, a geometric active contours model, low resolution segmentation with elevation data and high resolution optimization. Compared with the conventional segmentation methods, our approach has the advantages of dealing with complicated satellite images, automatically initializing a number of snakes based on colour and texture features, accurately and rapidly identifying the target objects with elevation data.

4.7.1 Vector-valued Scanning Algorithm

This algorithm is used for automatically producing zero level curves.

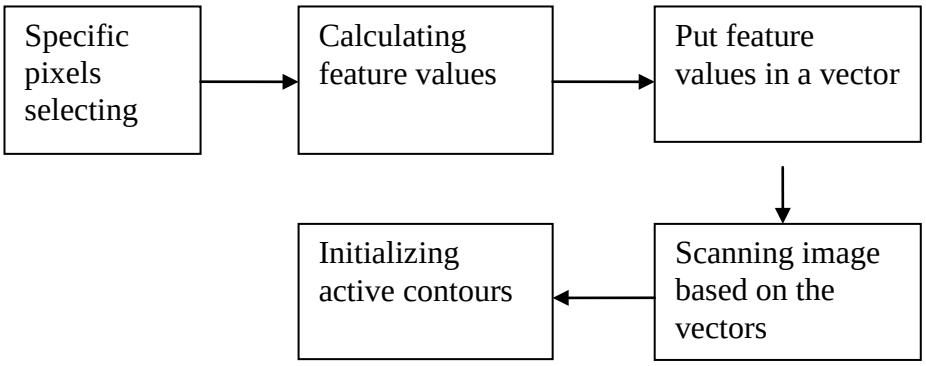


Figure.4.1. Vector-valued scanning algorithm

For an image $I(x, y)$ with the size of $s_1 * s_2$, the procedures of this algorithm can be described as follows (Figure 4.1): 1. First we select several specific pixels within the target object. 2. For each point $I(x_i, y_j)$, we take it as the center of a

rectangular cell and gives proper length (*l*) and width (*w*) of that cell. 3. We calculate interior pixels values of each cell in order to get the minimum, maximum, average and standard deviation values together with texture features. Then all of these values are compared among all cells for the purpose of acquiring appropriate image features. In particular, the specific image features such as color information, standard deviation values and terrain image gradient will be stored in a vector for image scanning purpose. 4. In image *I(x, y)*, we automatically assign initial snakes as if their specific values accord to the values inside that vector.

4.7.2 A Geometric Active Contours Model

In image segmentation, active contours are dynamic curves that move toward the object boundaries. The main advantages of active contours methods are the accuracy and the automatic handling of topological changes while the main drawbacks are computational cost and local minimum. However, our specific method has been proposed in order to speed up contours evolution in global region.

To achieve this goal, we explicitly define an external energy that can move the zero level curve towards the object boundaries. Let *I* be an image, and g be the edge indicator defined by

$$g = \frac{1}{1+\left|\nabla G_\sigma * I\right|^2} \tag{4.7}$$

where G_σ is the Gaussian kernel with standard deviation σ. We define an external energy for the contour C as below:

$$A_{g,p,q}(C) = pB_g(C) + qD_g(C) \tag{4.8}$$

where $p > 0$ and q is constant, and the terms $B_g(C)$ and $D_g(C)$ are defined by

$$B_g(C) = \int_\Omega g\delta(C)|\nabla C|didj \tag{4.9}$$

$$D_g(C) = \int_\Omega gH(-C)didj \tag{4.10}$$

Let $\Omega \subset R^2$ be the image domain, the δ is the Univariate Dirac Function, and H is the Heaviside Function.

Then we define the internal energy tR(C) in equation 4.11, where t >0 is a parameter controlling the effect of penalizing the deviation of C from a signed distance.

$$tR(C) = \int_\Omega \frac{1}{2}(|\nabla C| - 1)^2 didj \tag{4.11}$$

At last we define the total energy function for the geometric active contour model.

$$A(C) = tR(C) + A_{g,p,q}(C) \tag{4.12}$$

The external energy $A_{g,p,q}$ drives the zero level set toward the object boundaries, while the internal energy $tR(C)$ penalizes the deviation of C from a signed distance function during its evolution.

The steepest descent process for minimization of the functional *A(C)* is the following gradient flow:

$$\frac{\partial C}{\partial t} = t\left[\Delta C - div\left(\frac{\nabla C}{|\nabla C|}\right)\right] + p\delta(C)div\left(g\frac{\nabla C}{|\nabla C|}\right) + qg\delta(C) \tag{4.13}$$

This gradient flow is the evolution equation of the level set function in the proposed method.

4.7.3 Low Resolution Segmentation with Elevation Data

As the active contour is defined, we want to integrate this model with elevation data for further image processing. This approach can be summarized in the following way (Figure 4.2): converting original image and elevation data into a greyscale image, setting different threshold values for the greyscale image in

order to get the coarse areas of target object, initializing snakes in the coarse areas based on vector-valued scanning algorithm. At last, our active contours model is used for curve evolution which will extract the accurate object boundary.

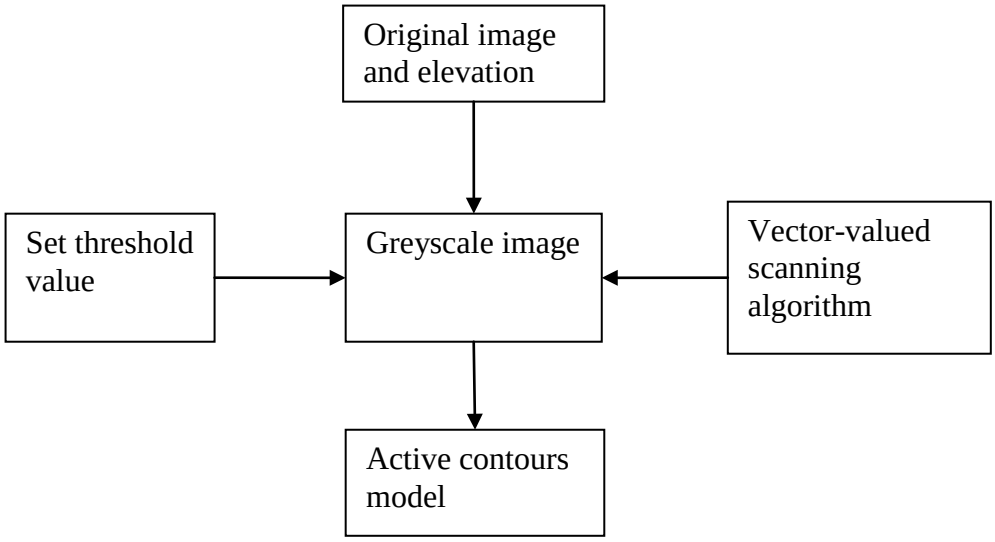


Figure 4.2 Active contour with elevation data

4.7.4 High Resolution Optimization

After achieving boundary at low resolution, the next step is passing extracted contour to high resolution for optimization. Resolution is the number of pixels in a linear inch (i.e. pixels per inch). The more pixels per inch, the higher the image resolution will be. More pixels create better image quality because you end up with more realistic representations of color, better gradations of both individual colors and gray tones, and crisper images in general.

The algorithm of contour passing is defined in formulation 4.14:

$$U_h(a_1 * x - i, a_1 * y - j) = U_l(x, y) \tag{4.14}$$

where a_1 is the scaling factor, U_l and U_h are represented for contour at low and high resolution. The iterative parameters *i* and *j* are separately from 0 to a_1. In Figure 4.3, we give an example of utilizing this method.

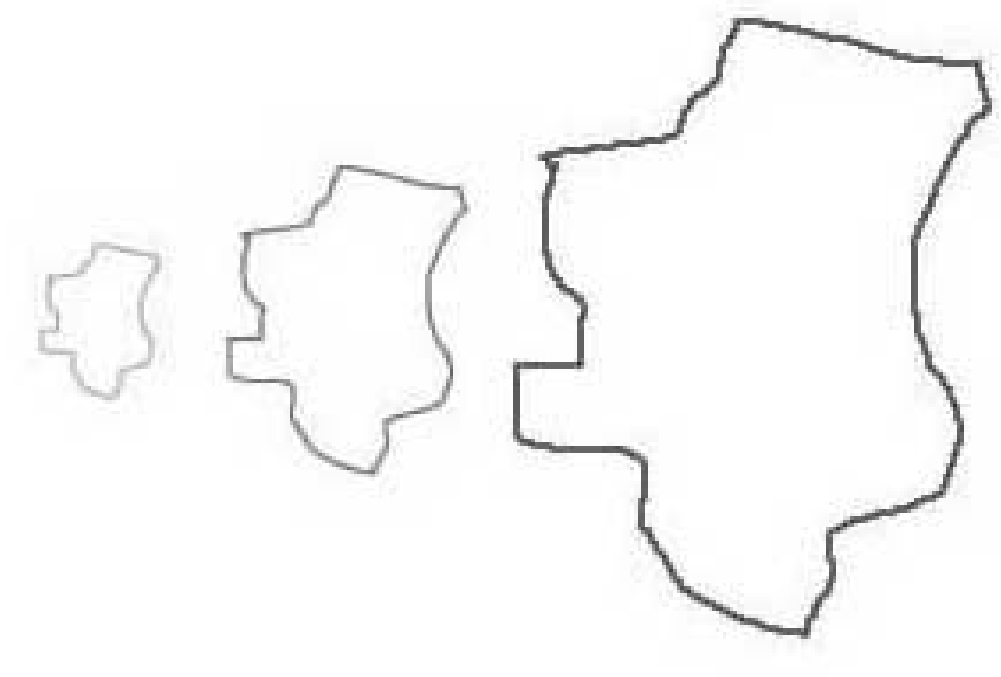


Figure 4.3 Contour from low to high resolution

For optimization, we begin with the domain of an image $I(x,y)$. Let u be the average intensity inside the curve U_h, and v outside the curve U_h. The basic idea is to minimize the following energy:

$$E = -\frac{1}{2}(u-v)^2 \tag{4.15}$$

The formulation 4.15 is the general formulation; we make use of this expression to compute the gradient descent curve evolution for E which is shown in formulation 4.16:

$$\frac{d\vec{C}}{dt} = -\nabla E = (u-v)\left(\frac{I-u}{A_u} + \frac{I-v}{A_v}\right)\vec{N} \tag{4.16}$$

where $\vec{N}$ denotes the outward unit normal of $\vec{C}$;

$A_u = \int_{R_u} dA$, R_u denotes the interior of U_h;

$A_v = \int_{R_v} dA$, R_v denotes the exterior of U_h.

4.7.5 Experiment Results

Satellite images can hardly be automatically segmented, due to their inherent complexity, low signal-to-noise ratio, undesired images features and other factors further complicate this issue. Manual and semi-automatic tracking of images are still the common methods for obtaining the segmentations of complicated images. We want to automatically and cost-effectively segment an image from our proposed method. This is the main point that motivates us. In our experiment, we select the city Limerick for our research object. The low resolution image of that city is shown in Figure 4.4 with the size of 300*200.

Figure 4.4 Low resolution image

Figure 4.5 represents the initial snakes generated from vector-valued scanning algorithm.

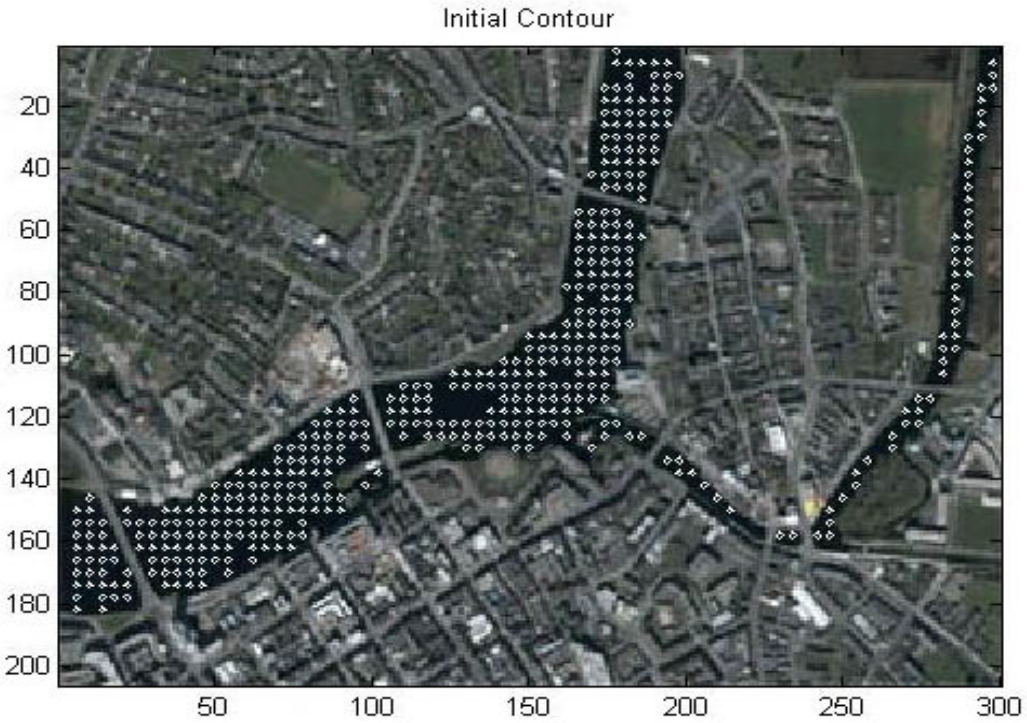


Figure 4.5 Initial snakes from vector-valued scanning algorithm

All of these initial snakes will move toward the river boundary. If one snake is getting close to the other one, then they will be automatically merged together for a big snake and so on.

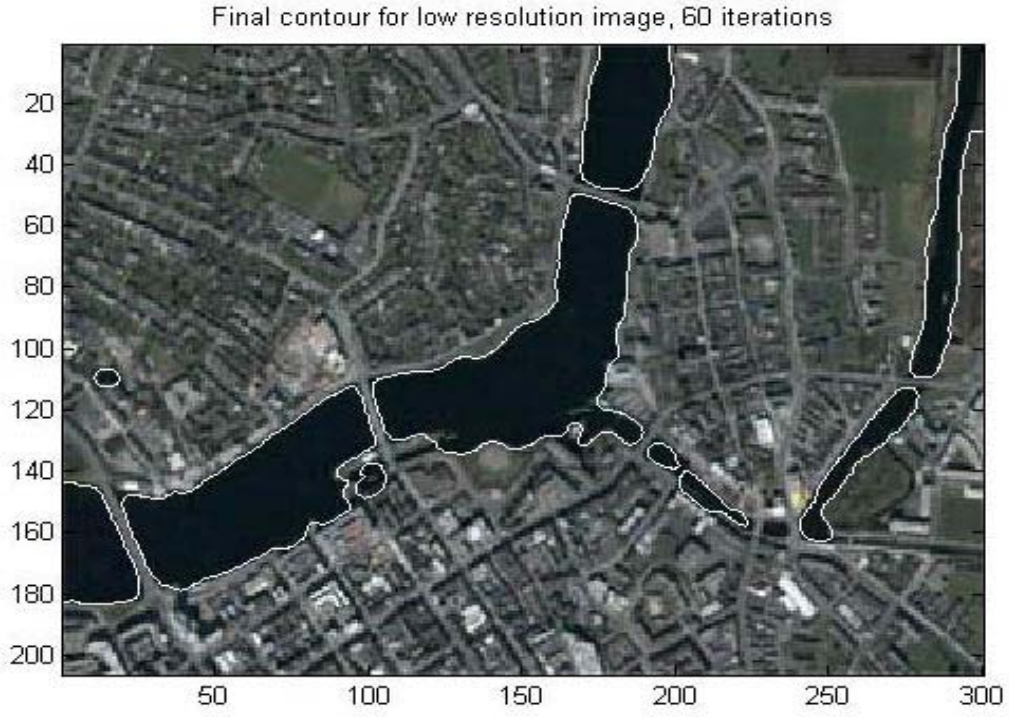


Figure 4.6 Final contours in low resolution image

The final snake in low resolution is shown in Figure 4.6 which is just after 60 iterations. At last, the river contour will be passed to high resolution for optimization.

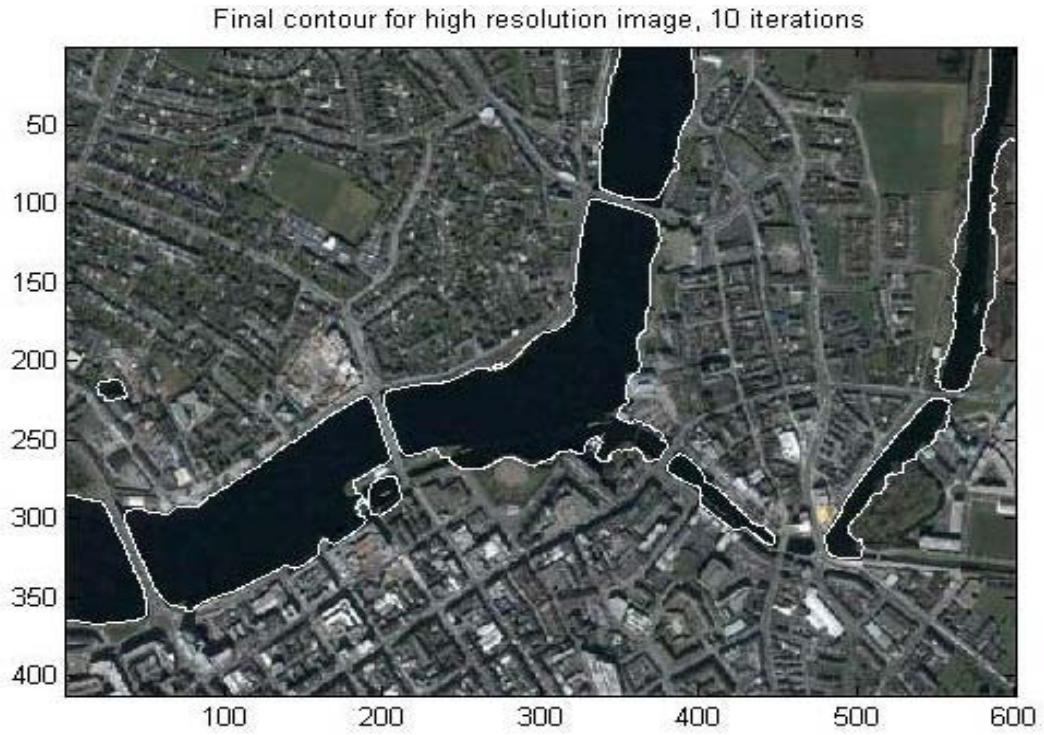


Figure 4.7 Final optimized contours in high resolution image

Figure 4.7 shows the final optimized contour after 10 iterations which is consistence with the river boundary. The experiment results show that our proposed segmentation method is fast and accurate.

During the monsoon season in 1998, a severe flooding occurred in the Yangtze River basin that was the worst event recorded since 1955 brought large economic losses in China. From this point of view, the second experiment has been conducted for recognizing one part of Yangtze River which is the first step trying to simulate and predict the flood over the Yangtze River Basin. The river boundary has been identified by active contour with white colour which is presented in Figure 4.10.

Figure 4.8 Initial snakes from vector-valued scanning algorithm

Figure 4.9 Final contours in low resolution image

Figure 4.10 Final optimized contours in high resolution image

The third experiment has been conducted for one urban area image segmentation. The urban image is captured from google earth, which is shown in Figure 4.11. Figure 4.12 shows the segmented urban image of forest, river, sea and residential regions with different colour.

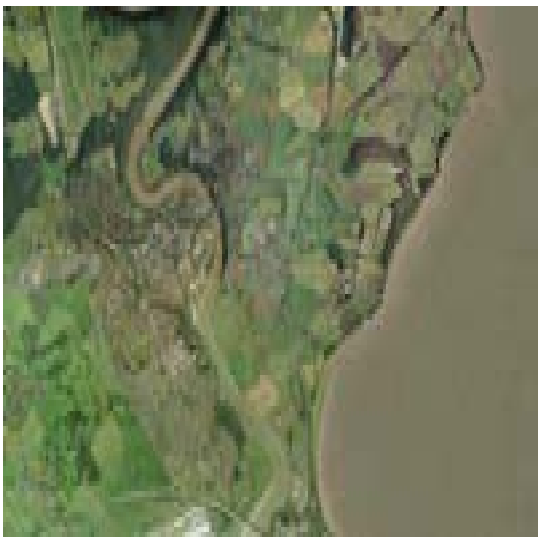


Figure 4.11 Urban image

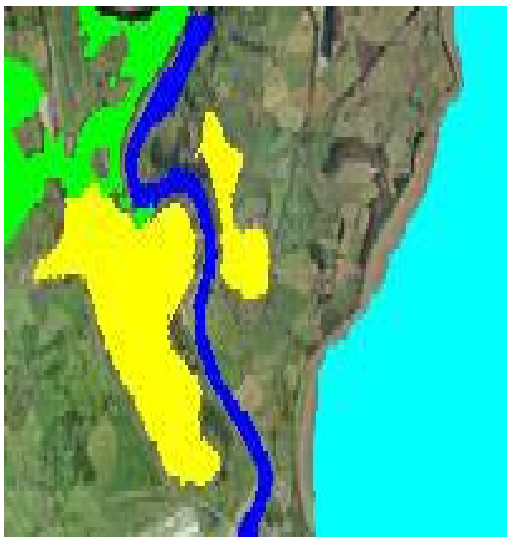


Figure 4.12 Segmented urban images

4.8 Environmental Objects Identification

In this part, environmental objects are identified from satellite images in meter resolution. At first, some basic types of primitive for environmental objects are defined. By using our geometric active contours model, the boundaries of environmental objects can be extracted and saved in binary maps. Then an intelligent shape match method is proposed which analysis the binary maps and choose the most similar types of primitives to model the environmental objects.

4.8.1 The Basic Types of Primitive

A simple environmental object is taken into account and modelled as a cuboid, and the spatial distribution of the building objects is assumed not crowded, i.e., without serious shadowing and superposition. The cubical object is described by seven geometric parameters, C(θ,X,Y,Z,L,H,D): 3D position, 3D size and orientation.

Figure 4.13 shows the cubical projection on the plane for primitive types of buildings. It can be seen that these two types of primitive are able to compose a number of combinations of primitive types which are shown in Figure 4.14. There are twelve types and will be used in the shape match method for building reconstruction in the following section.

Figure 4.13 Primitive types

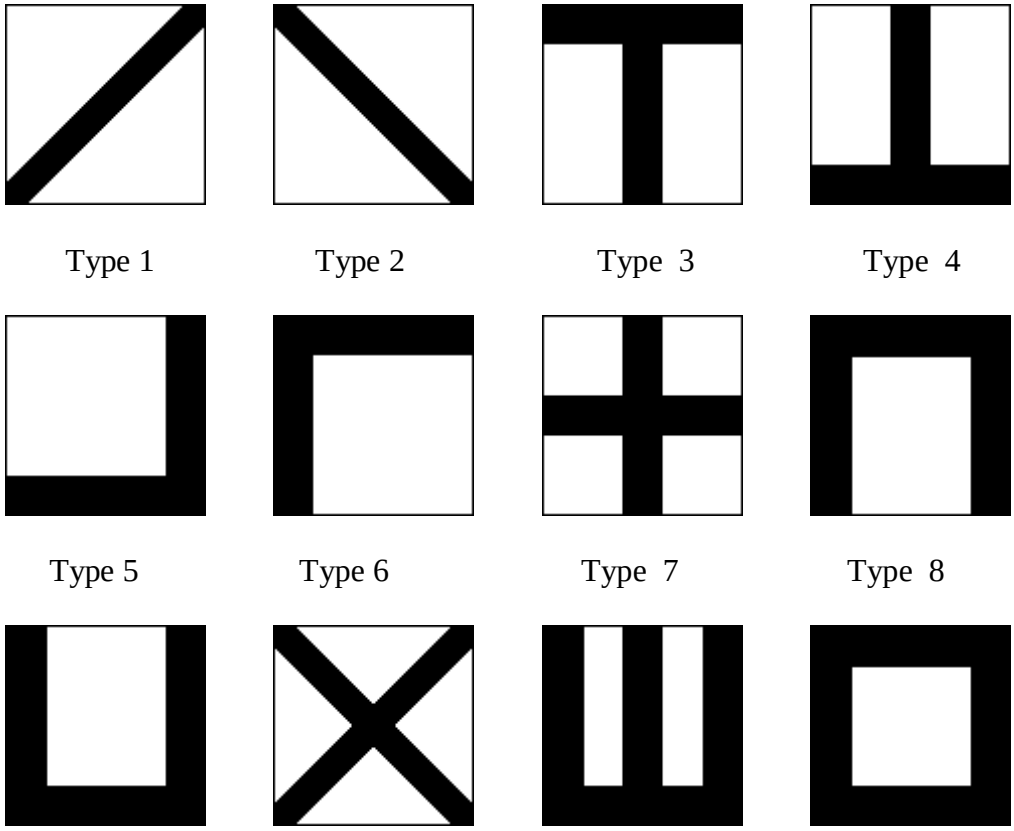


Figure 4.14 Combinations of primitive types

4.8.2 Environmental Objects Detection

Although we have defined the primitive types, the location of each environmental object is still unknown. By means of our geometric active contours model, we have successfully extracted boundaries of every environmental object which is shown in Figure 4.16.

Figure 4.15 Street, river and bulding map

Figure 4.16 Extracted buildings

4.8.3 Intelligent Shape Match Method

Shape match method is widely used through template in the spatial domain. Template operation can be expressed as prototype matching and could be very useful as an elementary image filter [Schalkoff 1989]. A template is an array of values, which covers a local area of an image at each time. It can be placed frame by frame or pixel by pixel, sequentially over the whole image. At each step, the elements of the template, i.e. the array of values, will be multiplied by the pixel grey levels of the selection of the image which corresponds exactly in size to the template position at that particular step. The sum of the products will give an assessment of the degree to which the image matches the template.

One of templates for primitive feature extraction is designed as a set of 10*10 size masks as shown in Figure 4.17. The extracted buildings image needs to be normalized (i.e. whole image size = (10*P)*(10*Q), P, Q=1, 2, 3……..) for the purpose of corresponding to the size of the primitive.

-1	-1	-1	-1	10	10	-1	-1	-1	-1
-1	-1	-1	-1	10	10	-1	-1	-1	-1
-1	-1	-1	-1	10	10	-1	-1	-1	-1
-1	-1	-1	-1	10	10	-1	-1	-1	-1
-1	-1	-1	-1	10	10	-1	-1	-1	-1
-1	-1	-1	-1	10	10	-1	-1	-1	-1
-1	-1	-1	-1	10	10	-1	-1	-1	-1
-1	-1	-1	-1	10	10	-1	-1	-1	-1
-1	-1	-1	-1	10	10	-1	-1	-1	-1
-1	-1	-1	-1	10	10	-1	-1	-1	-1

Figure 4.17 One template used in the measurement system

Let $W_k(i,j) = [w_{ij}^k]$, i,j=1,2, …,10 and k=1,2, …, 12 be twelve dimensional vectors formed from the above templates and $I_{(i,j)} = [I_{ij}]$, i,j= 1,2,…,10 be a vector formed from an equivalent array of the pixels of the local subimages:

I00	I01	I02	I03	I04	I05	I06	I07	I08	I09
I10	I11	I12	I13	I14	I15	I16	I17	I18	I19
I20	I21	I22	I23	I24	I25	I26	I27	I28	I29
I30	I31	I32	I33	I34	I35	I36	I37	I38	I39
I40	I41	I42	I43	I44	I45	I46	I47	I48	I49
I50	I51	I52	I53	I54	I55	I56	I57	I58	I59
I60	I61	I62	I63	I64	I65	I66	I67	I68	I69
I70	I71	I72	I73	I74	I75	I76	I77	I78	I79
I80	I81	I82	I83	I84	I85	I86	I87	I88	I89
I90	I91	I92	I93	I94	I95	I96	I97	I98	I99

The inner product of $W_k(i,j)$ and $I(i,j)$ is defined by

$$W_k(i,j) * I(i,j) = \sum_i \sum_j w_{kij} \cdot I_{ij} \quad (4.17)$$

In terms of extracting a particular primitive, the primitive which registers by the above procedure (i.e. the best of twelve possibilities in the local spatial region) can be taken as a potential primitive and is called a primitive candidate. A method for determining whether a primitive candidate is a real primitive is proposed. This involves the determination of the particular contrasts between the region of the candidate primitive detected and its immediate neighbouring background region in a local image. An example of region partition of the local subimage examined is shown in Figure 4.18, where a candidate primitive of type four is detected.

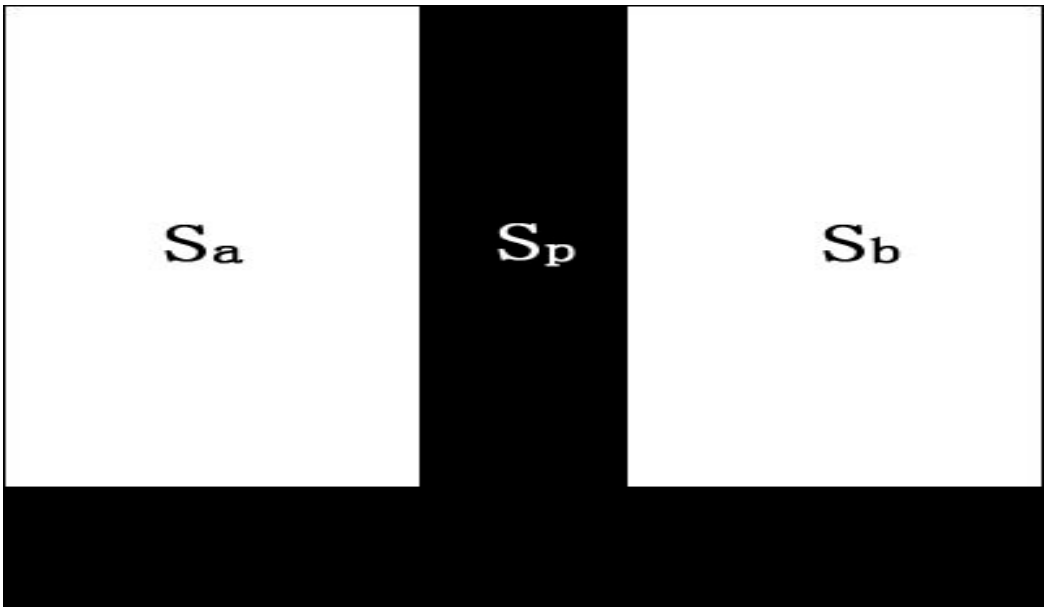


Figure 4.18 An example of a local image F(x,y); S_p: a primitive region P (type 4); S_a, S_b: the two neighbouring regions.

Suppose that the best match in the local area is determined. A primitive of the *m*th type is then registered as a candidate. We divide the 10*10 image F(x,y) examined into three regions: S_p, S_a, S_b, with one (S_p) in which the elements of W_k are positive and two (S_a,S_b) in which the elements of W_k are negative, for example, as shown in Figure 4.18. We then calculate the means of the intensity

values of pixels in these three regions. Let these be $\overline{F_p}(x,y)$, $\overline{F_a}(x,y)$, $\overline{F_b}(x,y)$. The candidate primitive is flagged as a primitive provided that the following conditions are met:

(i) $$\overline{F_p}(x,y) > \overline{F_a}(x,y) \tag{4.18}$$

(ii) $$\overline{F_p}(x,y) > \overline{F_b}(x,y) \tag{4.19}$$

(iii) $$\sigma_L^2(x,y) > TL \tag{4.20}$$

Where $\overline{F_p}(x,y)$ is the average intensity value of the candidate primitive pixels in S_p, and $\overline{F_a}(x,y)$ and $\overline{F_b}(x,y)$ are the average intensity values of the pixels of the two neighbouring background regions: S_a and S_b. The parameter $\sigma_L^2(x,y)$ is the local variance of the intensity values of pixels in the current area being examined, and *TL* is the preassigned threshold value for the variance. If anyone or more of the above conditions are not satisfied, the region or local area is flagged as having no primitive.

The local variance $\sigma_L^2(x,y)$ can be estimated as follows:

(i) The local mean value $\mu_L(x,y)$ of the grey levels of a local area is:

$$\mu_L(x,y) = \frac{1}{(2m+1)^2}\sum_{i=1}^{2m+1}\sum_{j=1}^{2m+1} F_{x+i,y+j} \tag{4.21}$$

Where, in the present application, m=4.

(ii) The local variance is then determined by:

$$\sigma_L^2(x,y) = \frac{1}{(2m+1)^2}\sum_{i=1}^{2m+1}\sum_{j=1}^{2m+1}[F_{x+i,y+j} - \mu_L(x,y)]^2 \tag{4.22}$$

The above example is just corresponding to primitive type 4. When compared with other primitive types with more neighbouring regions (i.e. S_c, S_d…), new conditions functions need to be added (i.e. $\overline{F_p}(x,y) > \overline{F_c}(x,y)$, $\overline{F_p}(x,y) > \overline{F_d}(x,y)\ldots$).

4.9 Conclusion

This chapter introduces several images processing techniques and reviews some typical image segmentation methods. After that, we propose a novel approach for rapidly and cost-effectively images segmentation. This method is represented by our multi-resolution framework which includes vector-valued scanning algorithm, a geometric active model, low resolution segmentation with elevation data and high resolution optimization. The multi-resolution method applied to active contour model speeds up the process and improves the final result. Environmental objects can be identified rapidly by intelligent shape match method which is useful for finding primitives which contain similar parts and counting the number of similar portions automatically. Further works will be focused on integrating this multi-resolution segmentation technique into 3D urban environment reconstruction.

Chapter 5 3D Environmental Reconstruction

3D environmental reconstruction is used for presenting a plentiful digital terrain and providing much more information which is integrated with our flood hazard warning system.

A prototype of 3D environmental reconstruction system has been developed in current work. This system is applied by using 3D virtual reality models, image segmentation, environmental objects identification, shape match method and intelligent reconstruction system. The image segmentation and environmental objects identification are described in previous chapters.

The 3D environmental reconstruction system offers the following advantages:

- It can automatically identify target areas based on our geometric active contours model and multi seed region growing algorithm.
- Environmental objects can be recreated with high speed by our shape match method in small-scale urban environment.
- It is able to reconstruct 3D large-scale environment intelligently based on specific segmented areas.

5.1 Introduction

Most of the existing research on environmental reconstruction has focused on creating aesthetically pleasing models for virtual tours and other entertainment purposes. Our purpose is to create models that can be used in applications such as emergency simulation, flood simulation, disaster preparedness, urban planning and change monitoring and environmental resource management. Such

applications require highly accurate models that are, therefore, usually created manually. The laborious process of hand digitizing and interactively crafting geometric models for buildings, trees, etc. is very time consuming.

In order to solve the problems, we propose to develop algorithms for automatically creating these models.

Our work has achieved two main objectives:

- Due to our geometric active contours model, multi seed region growing method and intelligent reconstruction system for forest area, sea area, urban area and river area, 3D large scale environment is reconstructed rapidly.
- For small scale environment, intelligent shape match method is used for detailed reconstruction which is much more accurate.

The chapter is structured as follows: Section 1 introduces the research on environmental reconstruction; Section 2 reviews automatic extraction of environmental objects such as building, vegetation and road; Section 3 explains our proposed method for 3D large-scale environment reconstruction and 3D small-scale detailed environment reconstruction; Section 4 draws the conclusion.

5.2 Automatic Extraction of Building, Vegetation and Road

5.2.1 Automatic Buildings Extraction

Accurate 3D city models are essential for a variety of applications, such as urban mapping and planning, mobile communications, environmental monitoring, risk assessment, and virtual tourism. Since manual extraction of urban 3D models is very costly and time consuming, these data are often either lacking or not up-to-date, while the demand for it continuously grows. Therefore, development

of automated algorithms capable of making the existing procedures more efficient is of great importance. Detection and description of buildings from aerial images have been an area of active research during the past two decades in both photogrammetric and computer vision communities. Nevertheless, the relatively slow progress in this domain only emphasizes the extreme difficulty of the task.

Not only do aerial images considerably vary with respect to their geometric and radiometric characteristics, but their content may be also very diverse and complex. It seems that only complex systems (multistep or multimethod) may succeed in achieving reliable results [Hsieh et al 1992]. Thus, the development of new methods improving and/or completing the existing ones may finally provide the ability to build operational systems. In this sense, we believe that the novel technique for building detection presented in this paper might be helpful.

There are very wide spectrum of approaches that exists for building extraction, based on analysis of digital elevation models (DEMs)[Baltsavias et al 1995] [Haala and Hanh 1995], and feature grouping, either in 2D [Collins et al 1995] or 3D [Roux and McKeown 1994] [Shufelt 1996], often including or followed by fitting either parametric [Lin et al 1994] [Lang and Forstner 1996] or generic models [Braun et al 1995] [Hendrickx et al 1997]. Good overviews of building extraction approaches are presented in [Gruen 1997] [Gruen and Navatia 1998]; a recent survey of models and strategies used for building extraction can be found in [Mayer 1999]. Thus, in this section we do not attempt to make a complete overview of the domain but rather to mention the approaches related to this work.

Building extraction can be subdivided into two interdependent tasks, detection and reconstruction, and it has been argued that reliable detection should

be supported by accurate reconstruction [Mayer 1999]. However, both tasks are very difficult by their own rights, especially in dense environments. While some recent approaches show the ability to accurately model quite complex building structures [Henricsson 1998] [Fischer et al 1998], an automatic initialization issue (i.e., detection and facet reconstruction) for model extraction within dense scenes is still to be addressed. One reasonable alternative is to use interactive initialization [Sahar and Krupnik 1999] [Nevatia and Noronha 1999], followed by fully automatic extraction. This approach is very practical, since a trained human operator can make the initialization rather fast, and in addition he or she supervises the extraction process, thus making it reliable. Nevertheless, the user-assisted methods remain time-consuming and therefore a need for fully automatic ones not only presents a challenging research issue, but also arises from the practical necessity. In this context, we suggest that the system should make first an effort of fully automatic extraction followed by a severe verification of the results. Then, it should ask for operator guidance only when the results are judged as uncertain. As system reliability can be considerably improved through using additional information sources, recent utilization of different cues, such as multiview stereo, range data, geometric constraints, colour, data-based knowledge, and context reasoning, in the process of building detection and reconstruction seems to be very promising [Noronha and Nevatia 1997]. The verification issue has been mainly addressed through the use of shadows [Liow and Pavlidis 1990], walls or vertical edges [Lin and Nevatia 1998], building microstructures (doors, windows) [Wang et al 1997], or constraint satisfaction [Fischer et al 1998]. Notice however that usually the existing methods deal with

suburban scenes, with rather simple-shaped and sparse structures, while building extraction in dense and complex urban scenes still remains a challenge.

5.2.2 Automatic Vegetation Extraction

Vegetation extraction from remote sensing imagery is the process of extracting vegetation information by interpreting satellite images based on the interpretation elements such as the image colour, texture, tone, pattern and association information, etc. Diverse methods have been developed to do this. Those methods can be broadly grouped either as supervised or as unsupervised depending on whether or not true ground data are inputted as references. General steps involved in vegetation mapping include image preprocessing and image classification.

Image preprocessing deals with all preparatory steps necessary to improve the quality of original images, which then results in the assignment of each pixel of the scene to one of the vegetation groups defined in a vegetation classification system or a membership matrix of the vegetation groups if fuzzy classification is adopted.

5.2.2.1 Preprocessing of Satellite Images

Preprocessing of satellite images prior to vegetation extraction is essential to remove noise and increase the interpretability of image data. This is particularly true when a time series of imagery is used or when an area is encompassed by many images since it is essentially important to make these images compatible spatially and spectrally. The ideal result of image preprocessing is that all images after preprocessing should appear as if they were acquired from the same sensor [Hall et al 1991]. Botanist and ecologist should keep in mind that while image

preprocessing is the prerequisite for vegetation extraction from remote sensing images, the preprocessing procedures listed below may not be always needed because some of these preprocessing procedures may have been done by image distribution agencies. Image preprocessing commonly comprises a series of operations, including but not limited to bad lines replacement, radiometric correction, geometric correction, image enhancement and masking (e.g. for clouds, water, irrelevant features) although variations may exist for images acquired by different sensors.

Bad line replacement is to determine the overall quality of the images (e.g. missing data lines) through visually previewing the images band by band. The visual review is usually done at full extents while attention is focused on identifying lines or blocks of missing data in each band for further repairing. Image line replacement is a procedure that fills in missing lines with the line above, below or with an average of the two.

Radiometric correction of remote sensing data normally involves the process of correcting radiometric errors or distortions of digital images to improve the fidelity of the brightness values. Factors such as seasonal phenology, ground conditions and atmospheric conditions can contribute to variability in multi temporal spectral responses that may have little to do with the remote sensed objects themselves [Song and Woodcock 2003]. It is mandatory to differentiate real changes from noises through radiometric correction in cases where the spectral signals are not sufficiently strong to minimize the effects of these complicating factors. Several methods are available to make radioactive corrections; some of them are based on complex mathematical models that describe the main interactions involved. However, the values of certain

parameters (i.e. the atmospheric composition) must be known before applying them. Other radiometric correction methods are based on the observations of reference targets (e.g. water or desert land) whose radiometry is known. Whatever radiometric correction methods are, they can be classified into two types: absolute and relative correction [Cohen et al 2003] [Coppin et al 2004] [Du et al 2002]. The absolute radiometric correction is aimed toward extracting the absolute reflectance of scene objects at the surface of the earth, requiring the input of simultaneous atmospheric properties and sensor calibration, which are difficult to acquire in many cases[Chen et al 2005] [Du et al 2002] [Song et al 2001] . On the other hand, the relative radiometric correction is aimed toward reducing atmospheric and other unexpected variations among multiple images by adjusting the radiometric properties of target images to match a base image [Hall et al 1991], which proves to be easier to apply. Schroeder et al [Schroeder et al 2006] and Chen et al [Chen et al 2005] extensively compared the effectiveness of the absolute radiometric correction methods (6S model, MDDV model and DOS model) and the relative radiometric correction methods (MAD model and PIF model) and illustrated the pros and cons of each model.

Geometric correction is aimed to avoid geometric distortions from a distorted image and is achieved by establishing the relationship between the image coordinate system and the geographic coordinate system using the calibration data of the sensor, the measured data of position and altitude and the ground control points. Therefore, geometric correction usually includes the selection of a map projection system and the coregistration of satellite image data with other data that are used as the calibration reference. The outcome of geometric correction should obtain an error within plus or minus one pixel of its

true position, which allows for accurate spatial assessments and measurements of the data generated from the satellite imagery. The first order transformation and the nearest neighbour resampling of the uncorrected imagery are among those popularly adopted methods in geometric correction. The first order transformation, also known as the linear transformation, applies the standard linear equation ($y = mx + b$) to the X and Y coordinates of the ground control points. The nearest neighbour resampling method uses the value of the closest pixel to assign to the output pixel value and thus transfers original data values without averaging them. Therefore, the extremes and subtleties of the data values are not lost [ERDAS 1999].

Sometimes the images will be more distinguishable for interpretation if image enhancement is performed, which is aimed to emphasize and sharpen particular image features (i.e. particular species of vegetation) for visualization purpose. The traditional image enhancement include grey scale conversion, histogram conversion, colour composition, colour conversion between red, green and blue (RGB) and hue saturation intensity transform (HSI), etc., which are usually applied to the image output for image interpretation. Shyu and Leou [Shyu and Leou 1998] explained the limitations of traditional image enhancement methods and proposed a genetic algorithm approach that was proved more effective than the traditional ones.

In mapping vegetation cover using remote sensing images, especially mapping over large regions, cloud imposes a big noise for identifying vegetation and thus has to be removed or masked. Jang et al [Jang et al 2006] proposed a neural network to detect cloud in SPOT VEGETATION images. Walton and

Morgan [Walton and Morgan 1998] used cloud-free space shuttle photograph to detect and remove (mask) unwanted cloud covers in Landsat TM scenes.

5.2.2.2 Image Classification

Image classification, in a broad sense, is defined as the process of extracting differentiated classes or themes (e.g. land use categories, vegetation species) from raw remotely sensed satellite data. Obviously this definition includes the preprocessing of images. We here simply refer to the process following the image preprocessing as image classification. Techniques for extracting vegetation from preprocessed images are grouped into two types: traditional and improved methods.

5.2.2.2.1 Traditional Methods

The traditional methods employ the classical image classification algorithms, e.g. K-mean and ISODATA for unsupervised classification or the maximum likelihood classification (MLC) for supervised classification. Unsupervised approach is often used in thematic mapping (including vegetation cover mapping) from imagery. It is easy to apply and widely available in image processing and statistical software packages [Langley et al 2001]. Two most frequently used methods are the K-mean and the ISODATA clustering algorithms. Both of these algorithms involve iterative procedures. In general, both of them assign an arbitrary initial cluster vector first. The second step classifies each pixel to the closest cluster. In the third step, the new cluster mean vectors are calculated based on all the pixels in one cluster. The second and third steps are repeated until the gap between the iteration is small enough (or smaller than a preset threshold). Unsupervised classification methods are purely relying on spectrally pixel-based

statistics and incorporate no prior knowledge of the characteristics of the themes being studied. The benefit of applying unsupervised classification methods is to automatically convert raw image data into useful information so long as higher classification accuracy is achieved Tso and Olsen [Tso and Olsen 2005]. Alternatively, rather than purely spectral, Tso and Olsen [Tso and Olsen 2005] incorporated both spectral and contextual information to build a fundamental framework for unsupervised classification, Hidden Markov Models, which showed improvements in both classification accuracy and visual qualities. Algorithms of unsupervised classification were investigated and compared with regard to their abilities to reproduce ground data in a complex area by Duda and Canty [Duda and Canty 2002]. Despite the ease of its application, one disadvantage of the unsupervised classification is that the classification process has to be repeated again if new data (samples) are added.

By contrast, a supervised classification method is learning an established classification from a training dataset, which contains the predictor variables measured in each sampling unit and assigns prior classes to the sampling units [Lenka and Milan 2005]. The supervised classification is to assign new sampling units to the priori classes. Thus, the addition of new data has no impact on the established standards of classification once the classifier has been set up. MLC classifier is usually regarded as a classic and most widely used supervised classification for satellite images resting on the statistical distribution pattern [Sohn and Rebello 2002][Xu et al 2005]. However, MLC shows less satisfactory successes since the MLC assumption that the data follow Gaussian distribution may not always be held in complex areas.

5.2.2.2.2 Improved Classifiers

It is very common that the same vegetation type on ground may have different spectral features in remote sensed images. Also, different vegetation types may possess similar spectra, which makes very hard to obtain accurate classification results either using the traditional unsupervised classification or supervised classification. Searching for improved classification methods is always a hot research topic. However, strictly speaking, all classification methods are derived from the traditional methods as aforementioned, which provide the basic principles and techniques for image classification. Thus, improved methods usually focus on and expand on specific techniques or spectral features, which can lead to better classification results and thus deserve special attention. Great progress has been made in developing more powerful classifiers to extract vegetation covers from remote sensing images. For example, Stuart et al [Stuart et al 2006] developed continuous classifications using Landsat data to distinguish variations within Neotropical savannas and to characterize the boundaries between savanna areas, the associated gallery forests, seasonally dry forests and wetland communities. They proved that continuous classifications were better than MLC classification especially in complex land cover areas.

Extensive field knowledge and auxiliary data may help improve classification accuracy. Studies have shown that classification accuracy can be greatly improved after applying expert knowledge (empirical rules) and ancillary data to extract thematic features (e.g. vegetation groups)[Gad and Kusky 2006][Shrestha and Zinck 2001]. In a regional scale vegetation classification conducted in the Amanos Mountains region of southern central Turkey using Landsat images, Domac and Suzen [Domac and Suzen 2006] incorporated

vegetation-related environmental variables and considerably improved classification accuracy when compared with the traditional MLC method. Under many circumstances, however, gathering specific knowledge is an enormous task and obtaining ancillary data is very costly. Therefore, the knowledge-based classifications are not universally applicable.

Sohn and Rebello [Sohn and Rebello 2002] developed supervised and unsupervised spectral angle classifiers (SAC), which take into account, the fact that the spectra of the same type of surface objects are approximately linearly scaled variations of one another due to the atmospheric and topographic effects. Those SAC helped identify the distances between pairs of signatures for classification and were successfully applied in biotic community and land cover classification [Sohn and Qi 2005]. The adoption of VI including the most widely used NDVI and its refined form EVI, which is another method to map vegetation using optical remote sensing devices [DeFries et al 1995]. The principle of applying NDVI in vegetation mapping is that vegetation is highly reflective in the near infrared and highly absorptive in the visible red. The contrast between these channels can be used as an indicator of the status of the vegetation. In other word, NDVI is a biophysical parameter that correlates with photosynthetic activity of vegetation. In addition to providing an indication of the 'greenness' of the vegetation [Wang and Tenhunen 2004], NDVI is also able to offer valuable information of the dynamic changes of specific vegetation species given that multiple-time images are analyzed. Therefore, NDVI is a good indicator to reflect periodically dynamic changes of vegetation groups [Geerken et al 2005]. Particular vegetation groups can be identified through their unique phenology, or dynamic signals of NDVI [Lenney et al 1996], which is also known as

"Multi-temporal Image Classification". Another approach to identify specific vegetation groups is to study time series VI. For example, Bagan et al [Bagan et al 2005] applied the combined EVI multi-dataset generated from 16-day interval MODIS data during the growing season of plants as input parameters to match the features of vegetation groups and to classify the images. The classification results were compared with those of the traditional MLC method and the accuracy of the former exceeded that of the latter.

5.2.3 Automatic Road Extraction

Roads, as one of the most important man-made objects, are subjects of great concern to be extracted automatically. There is a large body of related work on road extraction from aerial and satellite imagery. For an overview we will focus on recent work or work which employs similar data or techniques, e.g., classification, construction of networks, bridging of gaps, or snakes, as our approach. In [Fischler et al 1981] two types of operators are combined: the type I operator is very reliable but will very likely not find all features of interest, whereas the type II operator extracts almost all features of interest, but with a possibly large error rate. Starting with the reliable type I road parts, gaps are bridged based on the type II results employing F* search. Wiedemann et al [Wiedemann et al 1998] extract and evaluate road networks from MOMS-2P satellite imagery with a resolution of about 6 m employing global grouping. The basis of this approach is the Steger line operator [Steger 1998]. A framework for the extraction of multispectral lines and edges for the recognition of roads in SPOT or Landsat imagery is proposed in [Busch 1996]. The use of snakes for the detection of changes in road databases from SPOT and Landsat satellite imagery

is demonstrated in [Klang 1998]. Wallace et al [Wallace et al 2001] present an approach designed for a wide variety of imagery. It is based on an object-oriented database which allows the modelling and utilization of relations between roads as well as other objects. Recently, road extraction using statistical modelling in the form of point processes and Reversible Jump Markov Chain Monte Carlo was proposed by Stoica et al [Stoica et al 2004].

There are an increasing number of papers on road extraction from high resolution satellite imagery, particularly IKONOS. Dial et al [Dial et al 2001] gives an overview over the properties of the IKONOS sensor and presents a road extraction approach making use of the multispectral capabilities of the imagery. A system for road extraction from multispectral imagery based on fuzzy logic is proposed by Amini et al [Amini et al 2002] . Mena and Malpica [Mena and Malpica 2003] segment colour images using the Dempster-Shafer theory of evidence for the fusion of texture, to extract linear features. Peteri and Ranchin [Peteri and Ranchin 2003] employ a multi-resolution snake for the extraction of urban road networks given existing but imprecise GIS data. Doucette et al [Doucette et al 2001] present a semi-automatic approach that uses a pre-classified imagery to detect roads using the so called “self-organising road map” (SORM). In [Zhang and Couloigner 2004] a multi-resolution analysis approach based on wavelets, road junction detection, and grouping is proposed. Mohammadzadeh et al [Mohammadzadeh et al 2004] introduce an approach based on fuzzy logic and mathematical morphology.

5.3 Proposed Approach

5.3.1 Large-scale Environment Reconstruction

In large scale area, it has been a challenge topic to automatically extract environmental objects from images due to the high object density and scene complexity. Applying normal image processing methods could not achieve satisfied performance, especially for high resolution satellite images. For this reason, a novel environmental image segmentation framework has been proposed by integrating our geometric active contours model and multi seed region growing method. The geometric active contours model is used for moving within an image domain to capture desired image features with high speed. This model has already been discussed in chapter 4; we will talk about multi seed region growing method in the following.

5.3.1.1 Multi Seed Region Growing Method

The multi seed region growing method takes a set of seeds as input along with the image. The seeds mark each of the objects to be segmented. The regions are iteratively grown by comparing all unallocated neighbouring pixels to the regions. The difference between a pixel's intensity value and the region's mean, δ, is used as a measure of similarity. The pixel with the smallest difference measured this way is allocated to the respective region. This process continues until all pixels are allocated to a region.

This method needs one seed inside the region of interest (ROI). In our model, we applied it to multi seed domain which has been defined below:

$$P = \sum_{j=1}^{n} R\,(I\,, X_j\,, Y_j\,, t\,) \tag{5.1}$$

P : logical output image of region

I : input image

X_j : the X position of the seed point

Y_j : the Y position of the seed point

t : maximum intensity distance

Several interested points have been chosen and output their x and y values to our method function. The region is iteratively grown by comparing all unallocated neighbouring pixels to the region. The difference between a pixel's intensity value and the region's mean is used as a measure of similarity. The pixel with the smallest difference measured this way is allocated to the respective region. This process stops when the intensity difference between region mean and new pixel become larger than a certain threshold (t).

$$|I_m - I_n| > t \quad (5.2)$$

5.3.1.2 Intelligent Reconstruction System

Although the real world model has been constructed, the details of the environment like vegetation or buildings are still not enough for viewers. It is necessary to recreate the whole models based on specific features of the object with different algorithms. Sea region and river region are in the nature of same colour information, contour based regeneration is suitable for reconstruction. However, it is not efficient for constructing forest region based on contour method due to its complex topography. Triangular mesh method is proposed for this purpose which is useful for forest region reconstruction. Since we have already generated simple building models, these are certainly applied into the residential region reconstruction.

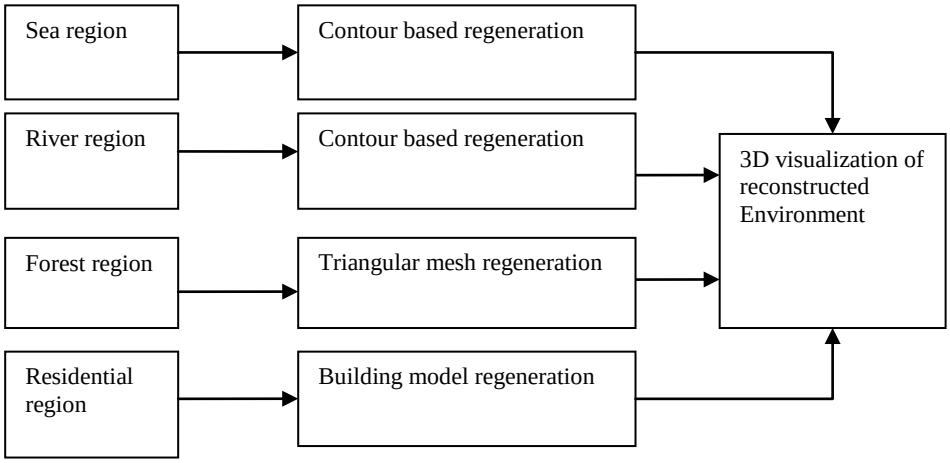


Figure 5.1 The structure of intelligent reconstruction system

Figure 5.1 sketches the workflow of the intelligent reconstruction system: i) Sea region: The method for sea region reconstruction is based on our active contours. Each contour is made up of curve points and will be automatically quad angulated in Maya. ii) River region: the method is as same as sea region regeneration. iii) Forest region: we use triangular mesh regeneration for reconstruction. Also provided with more detailed texture map, the forest area seems much more realistic. iv) Residential region: depends on position of the whole houses, we create all 3D building models with different sizes and angels in real environment.

5.3.1.3 Experiment Results

Large scale environment can be hardly generated and reconstructed fast and efficiently. We want to automatically and cost-effectively reconstruct the 3D real environment from our proposed method. In our experiment, we select the town Chepstow for our test object. The SRTM digital elevation date and texture map of that area have been prepared for input dataset. Figure.5.2 shows the 3D real environment of Chepstow generated by our digital terrain model method.

Figure.5.2 3D urban environment of Chepstow

Then the geometric active contours model and multi seed region growing method have been applied to the texture map of Chepstow for different areas segmentation. There are four identified areas listed below registered with distinct colour for sea, river, forest and residential regions.

Figure.5.3 Segmented sea area

Figure.5.4 Segmented river area

Figure.5.5 Segmented forest area

Figure.5.6 Segmented residential area

We automatically reconstruct the environment based on these segmented regions which is represented in Figure.5.7.

Figure 5.7 Reconstructed 3D environment of Chepstow

Figure 5.8 displays the close view of reconstructed 3D environment of Chepstow including sea model and detailed house models with different locations, orientations and sizes.

Figure 5.8 The close view of reconstructed 3D environment of Chepstow

5.3.2 Small-scale Detailed Environment Reconstruction

In this part, simple environmental objects are modelled as cuboids, and a novel approach for automatic reconstruction of 3D environmental objects from satellite images in meter resolution is developed. This approach consists of defining basic types of object primitive, environmental objects detection and intelligent shape match method which has been described in previous chapters. It is useful for finding primitives which contain similar parts and counting the number of similar portions automatically.

Two experiments have been carried out for validating our proposed environmental objects identification approach. The first experiment test has been conducted for reconstructing 3D small-scale detailed urban environment in one part of Manchester. Figure 5.9 shows the landform map of that area. The reconstructed building in one part of Manchester DTM has been shown in Figure 5.10. It is a satisfactory result with respect to the 3D ground.

Figure 5.9 Street, river and building map of one part of Manchester

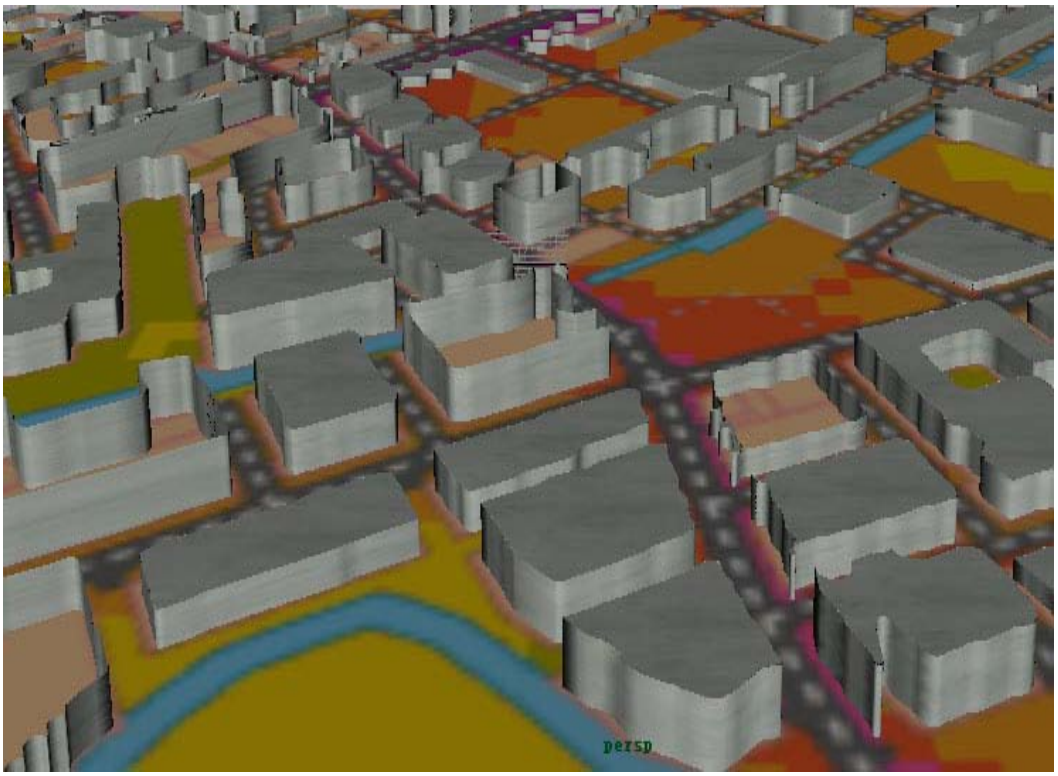


Figure 5.10 3D view of reconstructed buildings overlaid on Manchester area

The next experiment test was conducted for reconstructing one part of Leeds area. By means of our Geometric Active Contours Model and Multi Seed Region Growing Method, we have successfully extracted boundaries of every building from satellite images of that region. Then through our intelligent shape match method, the 3D small-scale detailed urban environment in one part of Leeds area has been successfully reconstructed. Figure 5.12 and Figure 5.13 are represented separately as far-distance and close-distance 3D view of the one reconstructed part of Leeds area.

Figure 5.11 Satellite image of one part of Leeds area

Figure 5.12 3D view of reconstructed buildings overlaid on Leeds area

Figure 5.13 3D close distance view of reconstructed buildings overlaid on Leeds area

5.4 Conclusion

This chapter proposes a 3D reconstruction approach for large-scale and small-scale detailed environment for different applications. 3D large-scale environmental reconstruction method is composed of our 3D digital real environment generation algorithm, environmental image segmentation by using geometric active contours model and multi seed region growing method, intelligent reconstruction system. 3D small-scale detailed environment reconstruction approach includes basic types of primitive, detailed buildings detection and intelligent shape match method. The experiment results indicate our method is capable of dealing with complicated and high resolution region reconstruction which is useful for many applications. In the next chapter, we will focus on 3D real time flood simulation as one of its important applications.

Chapter 6 Development of A Flood Warning Simulation System

This chapter describes the fundamental architecture of the flood warning simulation system, embodied in its components, their relationships to each other and the environment, as well as the principles that are governing its design.

This simulation system consists of three sub systems. For 3D real world modelling, NTSP algorithm and pyramidal data arrangement structure are proposed together which are able to deal with massive GIS terrain data and are capable of processing detailed terrain information with high resolution data set. For environmental quality between large area and detailed level, 3D environment reconstruction system is proposed by using 3D virtual reality models, image segmentation for object identification, shape match method and intelligent reconstruction system. Regarding to 3D flood simulation system, two new flooding models have been developed: flood spreading model and finite volume hydrodynamic model. Both flooding models are capable of realistically visualising the flooding in geometrically real 3D environments and can handle dynamic flood behaviour with fast speed and great accuracy in real time.

6.1 Introduction

The flood hazard warning system describes and defines the relationship between the different parts of the simulation system in order to offer not only numeric data or figures, but also more meaningful and appealing 3D visual information. Consequently, the performance of this simulation system depends on

the quality of the three sub systems: 3D real world modelling system, 3D environment reconstruction system and 3D flood simulation system.

The study achieved two main objectives: implementing a useful flood simulation with real world model and reconstructed environment for flood hazard warning; producing a friendly simulation system interface for either a decision maker or experienced user.

In this chapter, we will discuss three sub systems of our simulation scheme which have been listed below:

- 3D real world modelling system: This is the initial step toward modelling the simulation environments. The ROAM algorithm for single terrain modelling and the NineTreeNodes algorithm for dynamic terrain modelling are combined together. The user interface developing platform was built using C/C++/OpenGL.
- 3D environment reconstruction system: In order to present clear and detailed environment, some dense areas of 3D real world needs to be reconstructed. Object segmentation from satellite images and shape match for intelligent reconstruction from segmented objects are two important components for achieving this goal. The user interface developing platform of 3D environment reconstruction system was built using Matlab and Maya.
- 3D flood simulation system: To predict accurate flood hazard, flood region spreading algorithm by level of water calculation, flood region search and merge process with flood area contours and modelling floods in a dense urban area using 2D shallow water

equations are applied in the flood simulation system. The user interface developing platform was built using Maya Mel.

This chapter is organized as follows: Section 1 gives an introduction of this simulation system architecture; Section 2 talks about the flowchart of the whole simulation system design. Section 3 discusses the 3D real world modelling system; Section 4 describes 3D urban environment construction system; Section 5 explains 3D flood simulation system; Section 6 draws the conclusion.

6.2 The Simulation System Design

This simulation platform is a centralized system. The architecture of the simulation platform is shown in Figure 6.1.

As shown in Figure 6.1, the interface is in charge of three components of this project input and visualization output. The input includes the 3D real world modelling and user instruction, 3D environment reconstruction and 3D flood simulation. The output is mainly the simulation result, which combines the three parts above. The 3D real world modelling is the basic step of the whole system that provides the terrain necessary for the simulation to be executed. At the beginning of simulation, the 3D real world enquires the elevation and image data via the interface and constructs the 3D real environments. During the modelling, 3D environment reconstruction system is added in for complicated and high resolution region reconstruction. Then, 3D flood simulation has been conducted based on the above two systems. It offers the flood region spreading algorithm, flood area contours and 2D shallow water equations to the simulation system and displays the simulation result via interface.

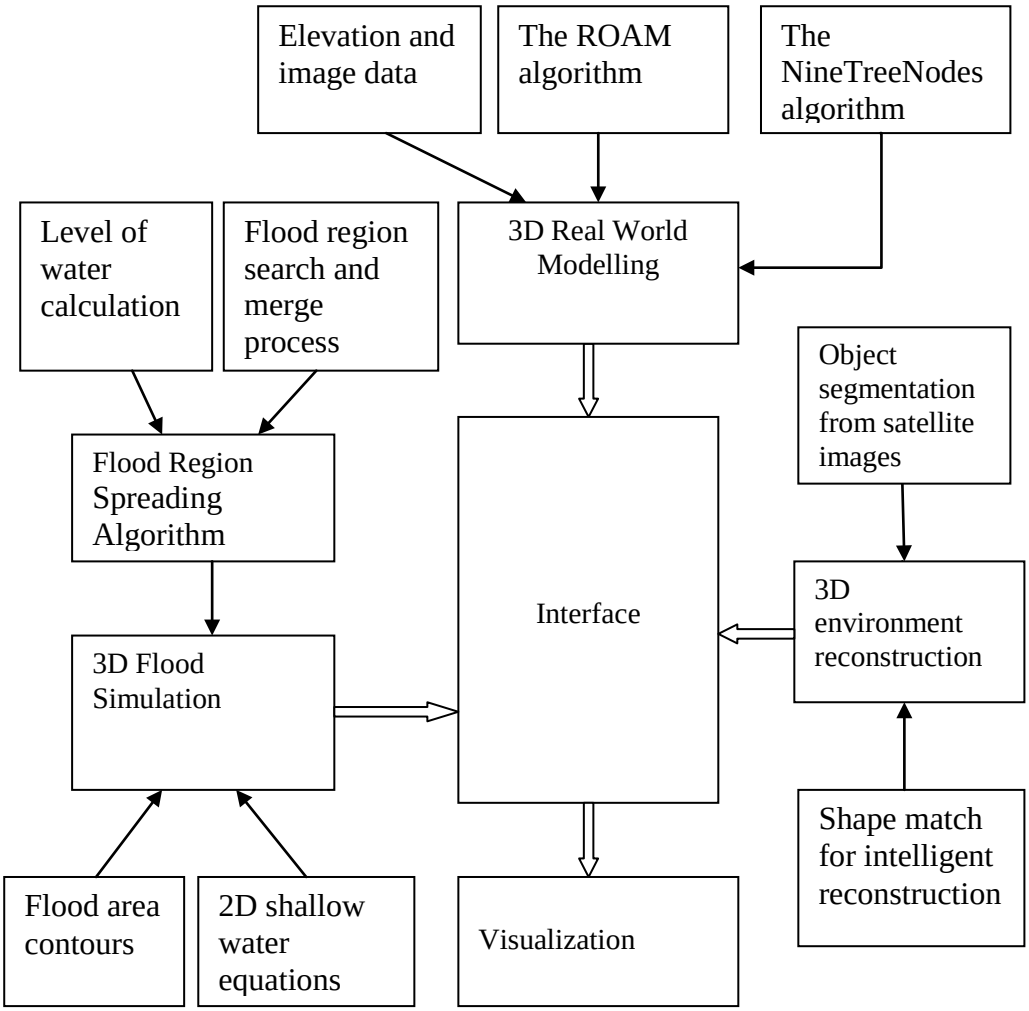


Figure 6.1 The architecture of the simulation platform.

The fundamental design approach of this simulation platform is modular functional decomposition, which is a common but efficient approach for software design and development. Modular software design refers simply to the structured and iterative decomposition of the system function into identifiable sub-functions. Each sub-function should be responsible for a single task and have a clearly defined interface to other components. Effective design minimizes the interconnections between functional elements. Fewer couplings between modules create the systems that are less complex and less prone to error. Software

structured in this manner is easier to test, and is more suitable for reuse from one project to the next.

This simulation system is tested on a desktop computer under windows XP operating system, the CPU is Pentium 4 (3.2 GHz) and the video card is Intel(R) 82915G/GV/910GL Express Chipset Family with 128MB memory. A number of simulation experiments have been designed to test this simulation platform, which include 3D real world modelling, 3D environment reconstruction and 3D flood simulation.

This simulation system is decomposed into three sub-systems and will be discussed in the following.

6.3 3D Real World Modelling System

Generating 3D real world is one of the major concerns while modelling the simulation environments. In order to achieve such a task in a quicker and easier way from raw data, an elevation grid data structure was used to contain the terrain information. The elevation grid is created in the x-y plane. All the grids have the same size on x-y plane and have a two dimensional coordinates corresponding to their location. Each grid square has an elevation value (height). The terrain information is saved in (.hgt) real elevation files.

3D real world environments are computer-generated environments that are often used to approximate the real world sceneries. An illustration of the 3D real world components is shown in Figure 6.2.

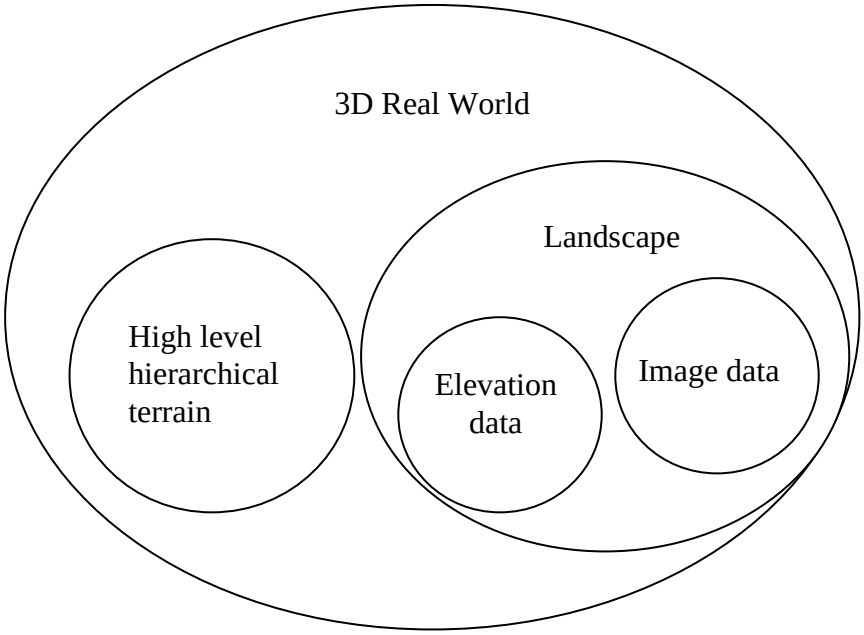


Figure 6.2 An illustration of the 3D real world components.

While constructing the 3d real world, firstly, the data of the environments is loaded from the external files, which include the terrain information and the image information; secondly the high level hierarchical terrain information is interpolated to form the detail elevation grid structure; at the end all the environmental information is combined together and the created terrain environments are displayed on the monitor screen. Figure 6.3 shows the process of constructing the 3D real world. This idea has been implemented in our 3D real world engines programmed by C++ includes main function and some specific classes. Figure 6.4 presents the structure of our 3D real world engines. The ROAM algorithm has been applied in terrain class, at the same time, NineTreeNodes algorithm for dynamic terrain modelling is carried out in NineTreeNodes class. Engine class is used for mouse device, keyboard device, font, texture, terrain and player initialization. Frustum class will allow us to create a terrain to keep track of our frustum. Player class is applied to control camera and observer's movement. Utils class is used to define all the controls parameters.

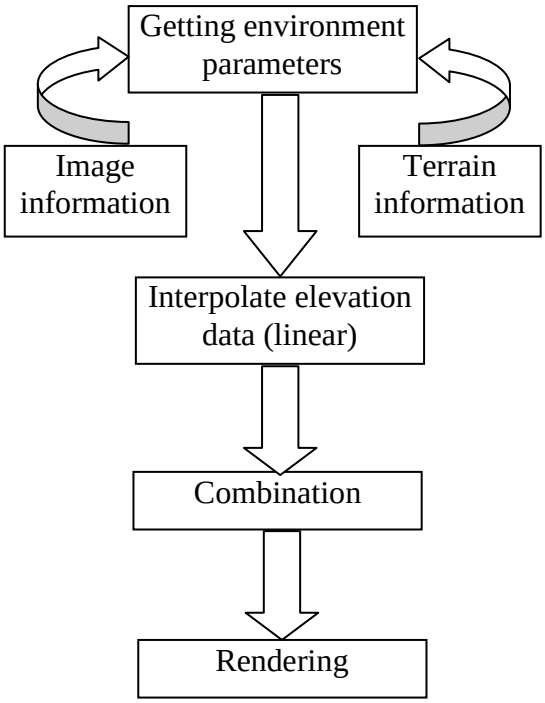


Figure 6.3 Construction of the 3D real world.

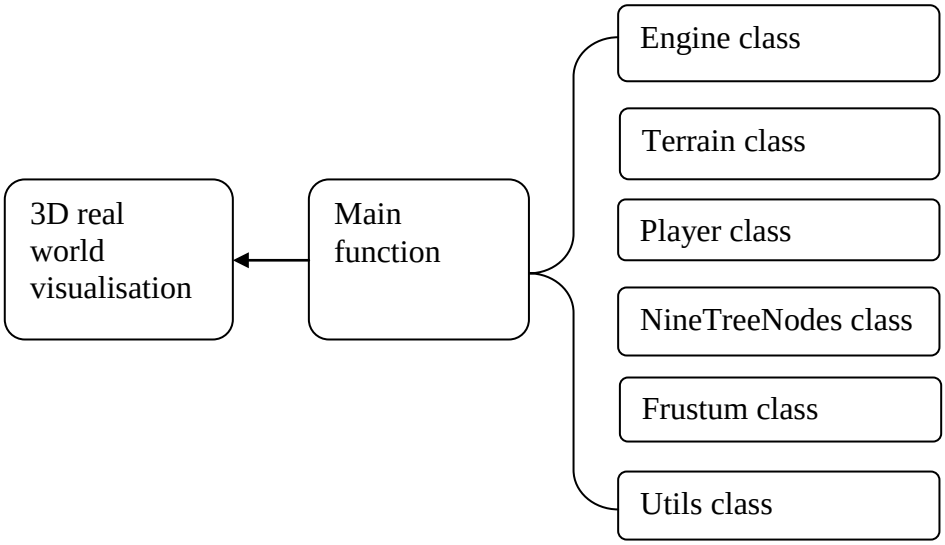


Figure 6.4 3D real world engines

In ROAM, frame coherence is achieved by using two queues called the split and merge queues. These queues represent the detail level state for a single bintree and solve the problem of cracks appearing between different level-of-detail algorithms that all LODs have to deal with.

In order to deal with large scale terrain modelling, the NineTreeNodes algorithm is proposed. It is made up of three major functions: InitTerrain( ), Mapmove( ), UpdateNode( ). Function InitTerrain( ) is used for loading elevation and image data, assigning parameters and generating nine terrains. Function Mapmove( ) is used for figuring out the direction of observer's movement. Function UpdateNode( ) is used for rearranging the nine terrains with new elevation and image data.

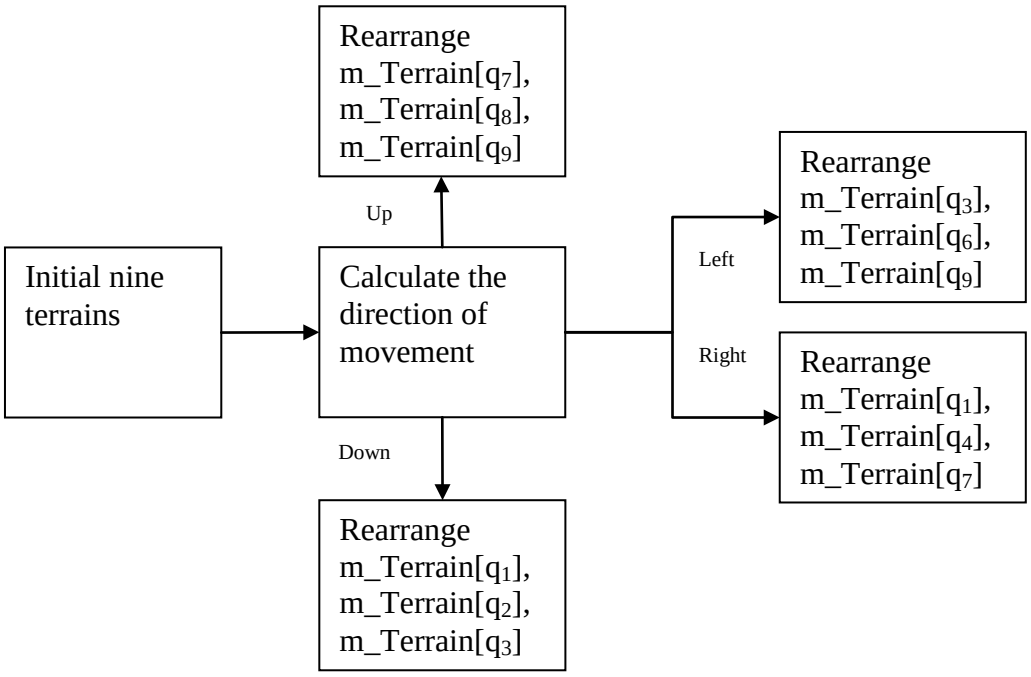


Figure 6.5 Diagram of NineTreeNodes algorithm

The pseudo code of the NineTreeNodes algorithm as shown in Figure 6.5 is listed below:

Initialize: terrain elevation data; terrain texture map;

Set column and row numbers; define centermap;

Begin

(1) Assign the observer's original position within the area of m_Terrain[q_5];

(2) Calculating the direction of observer's movement;

While playerposition!=null

Switch (direction)

Case up :

Calculate column and row number; Rearrange centermap;

Reload three terrains: m_Terrain[q_7], m_Terrain[q_8], m_Terrain[q_9];

Case down :

Calculate column and row number; Rearrange centermap;

Reload three terrains: m_Terrain[q_1], m_Terrain[q_2], m_Terrain[q_3];

Case left :

Calculate column and row number; Rearrange centermap;

Reload three terrains: m_Terrain[q_3], m_Terrain[q_6], m_Terrain[q_9];

Case right :

Calculate column and row number; Rearrange centermap;

Reload three terrains: m_Terrain[q_1], m_Terrain[q_4], m_Terrain[q_7];

End

This 3D real world modelling platform is developed using C/C++ programming language and OpenGL computer graphics library. The interface consists of a set of several hundred procedures and functions that allow a programmer to specify the objects and operations involved in producing high quality graphical images. A number of computer graphics techniques were utilized to improve the visual performance of the developed simulation platform. They are colour, texture mapping, lighting and viewpoints, etc.

Figure 6.6 shows a screenshot of the island skype 3D scenery which is a famous place in the UK. A commonly used format of digitally representing the 3D real world surface is the Digital Elevation Model (DEM). SRTM (.hgt) is a kind

of these files as we explain above. DEM consists of a sampled array of elevations for a number of ground positions at regularly spaced intervals to describe an axis-aligned grid of terrain. The distances between the sampled positions defines the resolution of a DEM and is the major determinant of its accuracy. The resolution of this DEM is 90m * 90m used for large-scale modelling. Constructed terrain geometry (Skye) is overlaid by aerial image corresponding to the same geographic area represented in the input DEM. From this test, only 9000 meshes have been used for modelling this real world, which proves that our approach is efficient and has a high-speed.

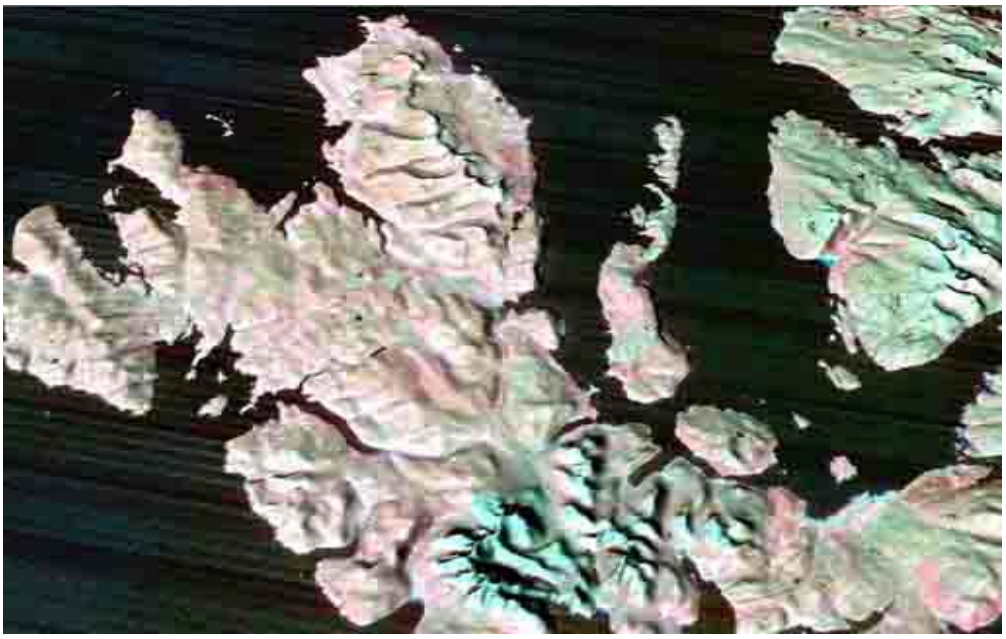


Figure 6.6 3D real world modelling.

6.4 3D Environment Reconstruction System

3D environment reconstruction system consists of image segmentation based on active contours and intelligent shape match reconstruction system. For image segmentation, it can be divided into four parts: (1) Forest area segmentation. (2) Sea area segmentation. (3) Urban area segmentation. (4) River area

segmentation. For shape match reconstruction, it can be divided into three parts: (1) Define primitive types with position, size and orientation geometric parameters. (2) Scan the whole image and get the possible object area for matching. (3) Assign the possible area with different primitive types. Figure 6.7 describes the diagram of 3D environment reconstruction system.

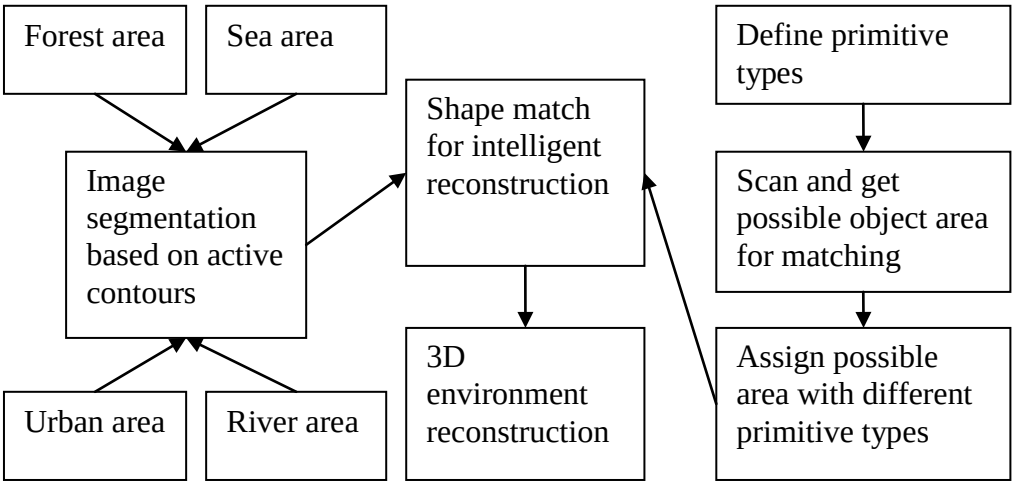


Figure 6.7 Diagram of 3D environment reconstruction system.

Figure 6.8 shows the flowchart of active contours segmentation. This is the initial step trying to reconstruct the 3D environment. The function select ( ) is used for choosing useful pixels and calculating average values, then the result will be saved in the vector R_f. The function search ( ) is used for assigning initial snakes by comparing searched values with selected values R_f.

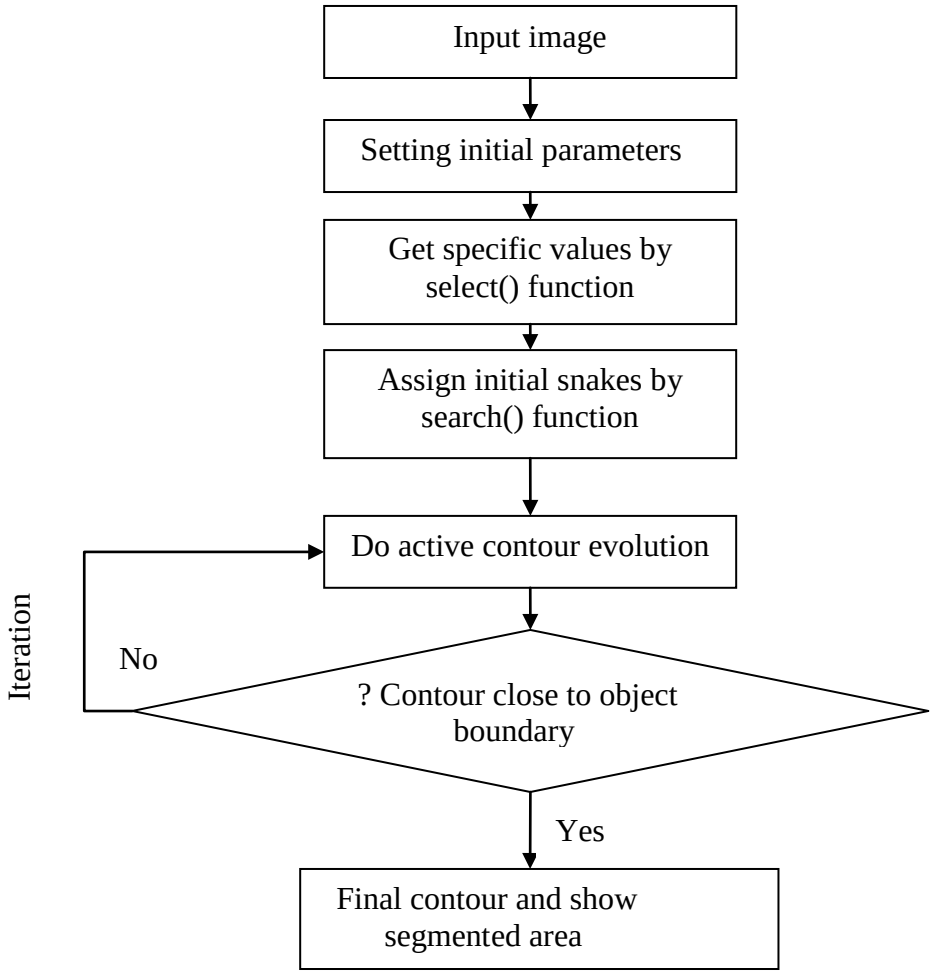


Figure 6.8 Image segmentation based on active contours

The pseudo code of the image segmentation based on active contours is listed below:

Initialize: Load in image data; Setting initial parameters; step=1;

(1) Choose useful pixels and calculate average values saved in R_f by select( );

(2) Assign initial snakes with search( ) by comparing searched values with selected values R_f

Begin

While contour != object boundary

(3) Contour evolution;

End

The pseudo code of sub function select ( ) is listed below:

While k < a1 (the numbers of how many center pixel we have chosen)

Begin

(1).Do interior pixels of each rectangular cell calculation;

(2).Minimum, maximum, average and standard deviation values computation saved in V_RGB;

(3).Get total minimum, maximum, average and standard deviation values from all rectangular cells and save in R_f.

End

The pseudo code of sub function search ( ) is listed below:

While I_{ij} (Image pixel) != null

Begin

(1).Get total minimum, maximum, average and standard deviation values from all rectangular cells and save in R1_f;

(2).Compare R1_f with R_f

If R1_f >R_f(min) and R1_f <R_f(max)

(3).Assign initial snakes;

(4).Active contours evolution and final contour;

End

An example of this algorithm has been shown from Figure 6.9 to Figure 6.11. At beginning, we automatically assign initial snakes as if their specific values are accord with the values inside the calculated vector. Then we do contours evolution by comparing the internal energy with external energy of the contours. When the internal energy is equal to the external energy, the evolution is finished. We finally get the final contour of the river area.

Figure 6.9 Initial active contours within river

Figure 6.10 The contours after several iterations

Figure 6.11 Final contours of river area

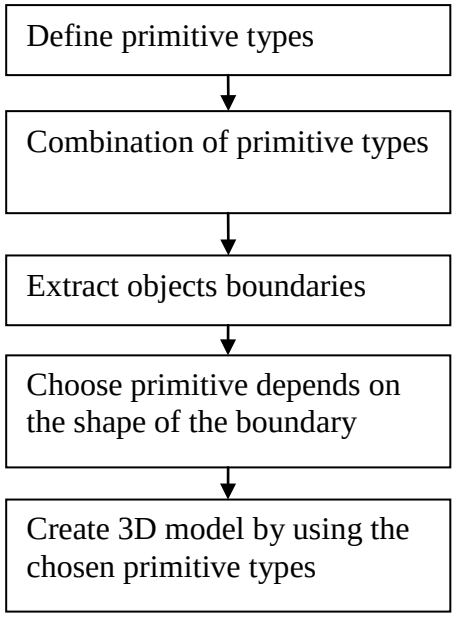


Figure 6.12 Intelligent shape match method

The pseudo code of the shape recognition in satellite images is listed below:

(1) Define primitive types: B_p1, B_p2;

(2) Combinations of B_p1, B_p2 and leading to other primitives from B_t1 to B_t12.

(3) Extrat objects' boundaries based on active contours.

(4) While contour == object boundary

Choose primitive (B_t1… B_t12);

Create 3d objects by using the chosen primitive types;

Figure 6.13 is a close distance view of the potential reconstructed flooding area. The resolution of this DEM is 10m * 10m and the texture map of this area is obtained from satellite image. The urban area, forest area, river area and sea area have been reconstructed automatically.

Figure 6.13 A close distance view of the potential reconstructed flooding area.

6.5 3D Flood Simulation System

3D flood simulation system consists of flood spreading algorithm and finite volume hydrodynamic model. For flood spreading algorithm, it can be divided into two parts: (1) Level of water calculation. (2) Flood region search and merge process. For finite volume hydrodynamic model, greyscale map and 2D shallow water equation are applied together for simulating flood inundation. Figure 6.14 describes the work flowchart of 3D flood simulation system.

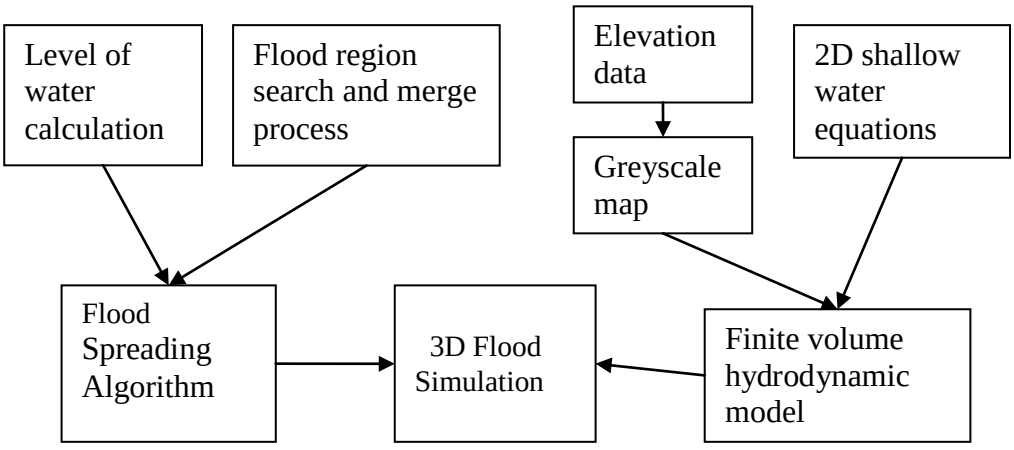


Figure 6.14 Diagram of 3D flood simulation

As the key component of 3D flood simulation, flood spreading algorithm is proposed for determining the flood spreading trend and calculating the flood inundation area. At first, a value of waterlevel is chosen which is a bit higher than initial cell elevation. Then, do search process from initial cell, put all connected cells into buffer B which meets the requirement of elevation lower than given waterlevel. After that, the flood regional volume V is calculated for comparison with flood storage capacity W_s on that area. While V is less than W_s, we need increase waterlevel to repeat above work. It will stop until V is equal to W_s. Figure 6.15 shows the process of flood spreading algorithm.

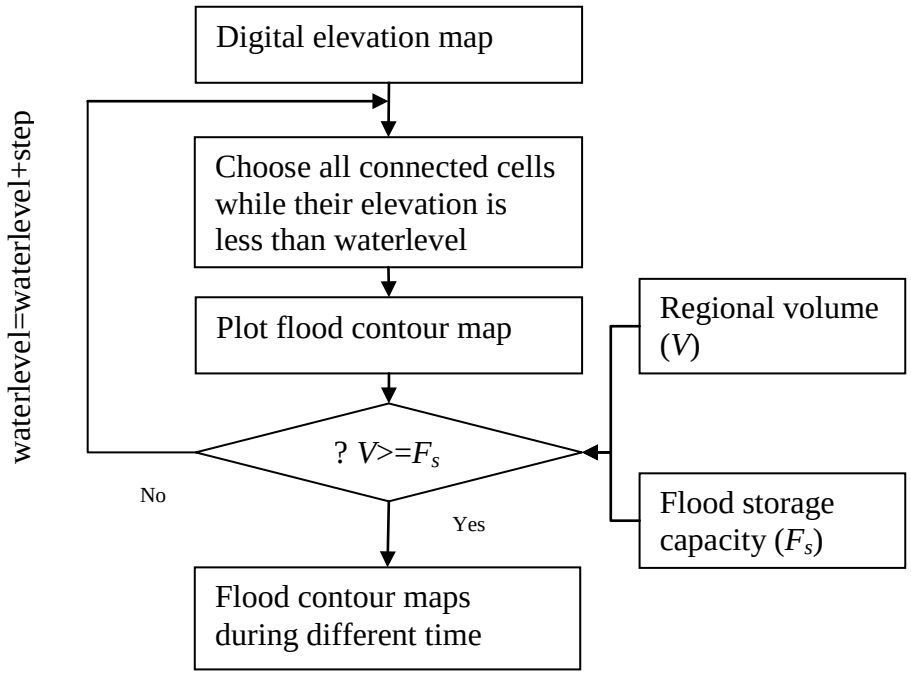


Figure 6.15 Flowchart of flood spreading algorithm

The pseudo code of the flood spreading algorithm is listed below:

(1) Waterlevel=Waterlevel + Step;

(2) While c != null

Begin

Select the identified cell c;

Obtain eight neighboring cell c_i with height h_i, i= 1 to 8;

If h_i < waterlevel;

Add c_i to buffer B;

End

Update identified cell c;

(3) Calculate flood regional volume (*V*):

V= [number of inundated cells] *[unit area of cell]*[sum of (waterlevel-h_i)];

Loop until $V>=W_s$;

(4) Flood contour maps during different time;

This 3D flood simulation system is developed using Matlab language and Maya embedded language. Matlab is a numerical computing environment and programming language. Our two flood models are defined and tested by using Matlab. Maya is a popular, integrated node-based 3D computer graphics and 3D modelling software. A cross-platform scripting language called Maya Embedded Language (MEL) is provided not only as a scripting language, but as means to customize Maya's core functionality. The 3D flood simulation interface is developed by using MEL. Additionally, numerical simulation results are transmitted and presented with digital terrain model in Maya environment which are very direct viewing.

Figure 6.16 is a close distance view of the flooding inundation. This flood simulation is implemented on the reconstructed environment which is shown in Figure 6.13. From the Figure 6.16, the terrible flood has submerged some low-lying areas with part of residential region and grassland; however, some high-lying areas such as forest regions are still safe. It is clear that our 3D flood

simulation system is suitable for dealing with high resolution detailed environment with precise result.

Figure 6.16 A close distance view of the flooding inundation.

6.6 Conclusion

This chapter discusses our simulation system architecture which includes 3D real world modelling, 3D environment reconstruction and 3D flood simulation systems. 3D real world modelling is composed of terrain and image data loading, the ROAM algorithm for single terrain modelling and the NineTreeNodes algorithm for dynamic terrain modelling. 3D environment reconstruction is made up of object segmentation based on active contours and intelligent shape match reconstruction. 3D flood simulation system consists of flood region spreading algorithm and simulating flood inundation by flood area contours. Our simulation platform offers convenient methods for constructing the real urban environments, specifying the flood hazard prevention tasks and monitoring the inundation areas.

Chapter 7 3D Flood Simulation

Prediction of inundation area helps to reduce flood-disaster damage efficiently by simulating the flood spreading procedure. In this chapter, two new general flood simulation models have been developed:

- Flood spreading model.
- The finite volume hydrodynamic model.

Flood spreading model is made up of four parts: (1) Flooding image spatial segmentation (2) Water level calculation process. (3) Standard gradient descent method for energy minimization. (4) Flood region search and merge process. The first model is built based on the flood contours energy minimization, standard gradient descent calculation in terrain elevation map applied into the model as well which provides a fast and an efficient flood prediction. The second model is built from 2D shallow water equations which can provide more detailed flood information and behaviour, and also can be used with the first model.

Both flood models utilise the digital terrain model which is built automatically from ROAM and NineTreeNodes algorithms mentioned in chapter 3. The simulation result provides supporting decisions information for flood disaster defence and basic data for flood hazard evaluation.

Our flood simulation system offers the following features:

- It is capable of realistically visualising the flooding in geometrically real 3D environments.
- It can handle dynamic flood behaviour with fast speed and great accuracy in real time.

7.1 Introduction

Large scale flooding is a serious natural disaster that frequently occurs in many parts of the world, in particular in America, Asia and Europe, where many countries have been at risk and people have suffered. The frequency, distribution and causes of floods over the last thirty years have been analyzed and reported by the Dartmouth flood observatory, and a five-fold increase in the number of floods per year has been observed since the 1980s [DFO]. In the UK, a 20 fold increase in the economic risk of flooding has been projected [Evans et al 2006]. Natural disasters such as Tsunami [USGS2004] and inland flooding cannot be prevented but damage can be reduced by proper planning. For this reason the simulation [Koshimura 2004] is essential for identifying areas likely to be affected with flood.

Simulation and modelling flood hazard warning are rapidly developing fields in hydrology [Boughton and Droop 2003]. Topographic data are crucial for flood inundation modelling and it is best to use recent and highly accurate topographic data. The flood simulation and model results are a good way of providing relevant information on how the flood is going to behave at the location where people live and how the flood will affect them. Current methods of flood hazard warning include numerical simulation (1D or 2D approach) [Mark et al 2004] [Marks and Bates 2000], remote sensing approaches [Wickland 1989] [Curran et al 1997] and GIS based flood simulation [Liu and Miao 2005]. Although these methods can solve some piratical problems, there are still two disadvantages of the approaches. One is the flood prevention system is constructed in 2D environments other than 3D real world. The other is that they forecast the flood based on the analysis of historical data.

In order to overcome the above problems, we propose a new dynamic flood simulation scheme which achieves two main objectives: (1). Flood module with 2D shallow water equations was activated for simulation of water flow in river reaches as well as river-bank overflow at flood plains. (2) Flood area contours which are extracted from each flood spreading progress are as input to our 3D reconstructed environment. According to the flood spreading time, these flood area contours are available for generating flood simulation plains. Benefit of our 3D flood simulation is to provide information for users such as land use planning, evacuation planning and environmental impact assessment.

In this chapter, we discuss previous flood simulation methods and propose our flood simulation approaches which have been listed below:

- Some flood simulation methods from 1D approach, 2D approach, combination of 1D and 2D method to finite volume approach are discussed. The basic performance and limitations of these methods are summarized.
- To solve existing problem, two new flood simulation models are proposed. In large-scale area, flood spreading model is implemented through flooding image spatial segmentation, water level calculation, standard gradient descent method for energy minimization and flood region search and merge process. For dense urban area, a finite volume hydrodynamic model is proposed based on 2D shallow water equations.

The chapter is structured as follows: Section 1 gives an introduction of flood simulation; Section 2 reviews some flood simulation methods from 1D approach, 2D approach, combination of 1D and 2D approach to finite volume approach; Section 3 introduces some terrain data used for constructing topography of the

river channel and the adjacent floodplain. Section 4 describes the flooding hazard warning system; Section 5 explains our flood spreading model which includes flooding image spatial segmentation, water level calculation, standard gradient descent method for energy minimization and flood region search and merges process. Section 6 talks about the finite volume hydrodynamic model based on 2D shallow water equations which can simulate the urban area flooding accurately; Section 7 makes the conclusion.

7.2 Flood Simulation Method

Flood simulation can be accomplished with one of several approaches. These could be organized by dimensionality, ranging from zero dimension models consisting of a water level versus flow rate rating curve, one dimensional (1D) models such as the well known HEC-RAS code, two-dimensional (2D) models that solve the shallow water equations, three-dimensional (3D) models that solve Navier–Stokes equations and hybrids that combine, for example, 1D and 2D approaches [Horritt and Bates 2002] [Lin et al 2006].

7.2.1 1D Approach

In the 1D approach, water flow is assumed to occur in one dominant spatial dimension aligned with the centre line of the main river channel. The geometry of the problem is represented in the model by channel and floodplain cross-sections perpendicular to the channel centreline. Measured distances between these cross-sections are also required by the computer model. Predictions of changes in water level and flow along the river are obtained from a numerical solution of the St. Venant equations [Barre 1871] of volume conservation.

$$\frac{\partial Q}{\partial x} + \frac{\partial A}{\partial t} = q \tag{7.1}$$

and momentum conservation:

$$\frac{\partial Q}{\partial t} + \frac{\partial}{\partial x}\left(\frac{\beta Q^2}{A}\right) + gA\frac{\partial H}{\partial x} - g\frac{AQ|Q|}{K^2} + q\frac{Q}{A}\cos\alpha = 0 \tag{7.2}$$

The equations model unsteady behaviour of the flood whilst taking into account the longitudinal hydrostatic pressure gradient, the frictional resistance of the bed and the momentum of any inflow. In flood inundations modelling these equations are normally solved using a finite difference technique.

In such models floodplain inundation is predicted as an integral part of the solution of the St. Venant equations, the approach is applicable where floodplains are of limited transverse dimension and are not separated from the main channel by embankments. The technique has the disadvantage that floodplain flow is assumed to be in one direction parallel to the main channel and that all transfer of momentum and mass is in the along channel direction. This is clearly inappropriate in cases where complex flow paths across the floodplain exist.

In the 1D+ approach, floodplains are modelled as storage reservoirs or floodplain storage cells (FSCs) with a horizontal water level over the storage cell surface. FSC geometry is defined using a water level vs. plan area relationship. Floodplain water level in the FSC is linked to the levels in the main channel using so-called spill units that model the exchange of water between the river and FSCs or between FSCs. Spill unit water exchanges between the main channel and FSCs or between FSCs can be estimated using discharge relationships based on a weir flow equation. Water level in each FSC is then computed using volume conservation. Unlike the 1D approach, 1D+ does not assume that flow is parallel

to the main channel flow; however, momentum is not conserved for the FSC computation. This allows instantaneous transfer of water through a FSC, which can lead to modelling problems in some circumstances. There are a number of commercially available river modelling packages that support the application of both the 1D and 1D+ approaches.

But 1D model has some limitations to include all details in modelling and it is very difficult to simulate local conditions on a small scale accurately [Mark et al 2004].

7.2.2 2D Approach

Models to predict flood inundation based on the 2D shallow water equations are classed here as 2D approaches and solve for water level and depth averaged velocities in two spatial dimensions. These can be expressed in the conservative form as:

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} = H \tag{7.3}$$

$$U = \begin{pmatrix} h \\ hu \\ hv \end{pmatrix}, \qquad F = \begin{pmatrix} hu \\ g\frac{h^2}{2} + hu^2 \\ huv \end{pmatrix} \tag{7.4}$$

$$G = \begin{pmatrix} hv \\ huv \\ g\frac{h^2}{2} + hv^2 \end{pmatrix}, H = \begin{pmatrix} 0 \\ gh(S_{0x} - S_{fx}) \\ gh(S_{0y} - S_{fy}) \end{pmatrix} \tag{7.5}$$

A solution to these equations can be obtained from a variety of numerical methods (e.g., finite difference, finite element or finite volume) and utilise different numerical grids (e.g., Cartesian or boundary fitted, structured or unstructured) all of which have advantages and disadvantages when it comes to floodplain modelling.

A further type of model is the raster-based inundation models, identified here as the 2D- approach. These have been developed specifically to take advantage of high resolution topographic data sets [Bates et al 2000]. Typically, channel flow is modelled using a 1D kinematic wave equation. During out of bank flow water is transferred to a 2D floodplain grid across which a 2D simulation is undertaken using a friction equation to compute flows between grid cells. The concept is similar to that adopted for the 1D+ approach, but with grid dimensions being considerably smaller than those of a typical FSC. As with the FSC approach momentum is not conserved for the 2D floodplain simulation.

For this study a 2D approach is adopted for a number of reasons. First, it is compatible with the native structure of geospatial elevation data which is mapped on a horizontal grid, compared with 1D approach which requires derivative products such as storage and conveyance tables. These in fact require a priori knowledge of the flow path and direction which can be difficult to ascertain in cases of severe flooding. Second, flood hydrodynamics can be truly multi-dimensional and in such cases its description warrants a multi-dimensional model formulation. Third, the 2D approach is justified based on the relative depth and breadth of floods and the appropriateness of the hydrostatic pressure approximation. Fourth, 2D flood inundation modelling has been an active area of research the past decade, particularly on the topic of Godunov type finite volume schemes, and today there are many robust simulation codes [Marks and Bates 2000] [Morris 2000].

The primary hypothesis of the 2D approach is that the vertical pressure distribution is hydrostatic [Katopodes and Strelkoff 1978], furthermore turbulent momentum dissipation can be modelled with bottom stresses according to a

quadratic drag law [Hogg and Pritchard 2004]. For this study, the drag coefficient was scaled by a Manning coefficient [Begnudelli and Sanders 2006].

7.2.3 Combination of 1D and 2D Approach

Over the past ten years significant advances have been made in integrating 1D and 2D models resulting in hydrodynamic model of floodplains and integrated 1D (channel flow) and 2D (overland flow) modelling such as DELFT-FLS [Sobek 2004]. The idea of integrating 1D hydrodynamic modelling technology, Digital Elevation Models and GIS systems is to take advantage of the best combination of 1D hydrodynamic data for rivers together with 2D terrain data, and presenting them in the GIS as maps [Horritt and Bates 2002].

The integrated one-dimensional and two-dimensional (1D and 2D) model development focuses on the extension of model capabilities in order to simulate flooding situations more accurately. This includes improving flood wave propagation over initially dry land, improving the presentation of hydraulic control (levees and embankment) in the floodplain and integration of one-dimensional hydraulic elements (pumps, bridges and regulator gates) [Horritt and Bates 2002]. The combined 1D and 2D modelling opens up the possibilities for studying flood control measures, flood forecasting, and development of flood evacuation plans.

7.2.4 Finite Volume Approach

Solution of the shallow-water equations has long been of interest to hydraulic engineers and over the last decade or so significant advances have come with the advent of Godunov-type finite volume schemes. Toro [Toro 2001] provides a review of many developments, and the accuracy of several dam-break algorithms is reviewed by Morris [Morris 2000]. The algorithm adopted in this

study, BreZo (FlowSimulation, LLC, Irvine, CA) is an operational version of the triangular grid algorithm reported by Begnudelli and Sanders [Begnudelli and Sanders 2006] and is similar to the quadrilateral grid algorithm reported by Bradford and Sanders[Bradford and Sanders 2002][Bradford and Sanders 2005] and Begnudelli and Sanders[Begnudelli and Sanders 2006]. BreZo does not account for infiltration of water into the soil or changes in topography resulting from flow induced erosion or sediment deposition. Grid generation for this study was accomplished using a Delaunay mesh generation tool [Shewchuk 1996]. Further, elevation at the vertices of the triangular grid was estimated from the DEM by linear interpolation. There are advantages to meshes that adapt based on topographic features [Bates et al 2003], but to compare DEM effects a common mesh should be used [Marks and Bates 2000].

7.3 Terrain Data

Both 1D and 2D flood simulation models require information on the topography of the river channel and the adjacent floodplain. In the case of urban floodplains, this includes the configuration and location of streets, buildings and other urban infrastructure. Since the mid-1990s, flood inundation models have made increasing use of high resolution and high accuracy digital terrain models (DTMs) based on Light Detection and Ranging (LiDAR) or airborne laser scanning data for this purpose.

7.3.1 LiDAR-based DEM

LiDAR is a modern technology in the field of remote sensing that can rapidly generate high-density, high-accuracy, geo-referenced digital elevation

data. LiDAR measurements are claimed to be subject to errors smaller than 0.15m vertically and 0.50m horizontally (root mean square errors). This is increasingly recognized as one of the few technologies (with, for example, photogrammetry) that can provide topographic data with suitable accuracy and resolution for applications in river and floodplain inundation modelling.

LiDAR is based on the combination of three different data collection tools: a laser scanner mounted on an aircraft; a Global Positioning System (GPS) providing the laser sensor position; and an Inertial Navigation System (INS) or Inertial Measurement Unit (IMU) providing the orientation of the aircraft (yaw, pitch and roll). Light pulses directed towards the ground are provided by the laser at a specific frequency (up to 100 kHz in state of the art systems). The sensor oscillates at a frequency and within a scan angle that can be set to provide the required swath width and data density (these also depend on the flight height). Each laser pulse (a decimeter-wide beam) is reflected by the ground or by objects above the ground. In some modern systems, multiple returns from the same laser pulse can be recorded (for example, high vegetation, low vegetation and ground). Each laser return provides a measured distance, which is then combined with the simultaneous INS and GPS measurements to provide a georeferenced ‘‘raw’’ LiDAR data point. Typical swath widths are between 0.25 and 1.5 h, where h is the flight height, normally between a few hundred and one thousand metres. At present, raw data density is around a few points per square metre but data collection at a much higher resolution is feasible and is becoming more common. It should be noticed that most LiDAR systems, which are currently available, do not penetrate water, so that LiDAR data are mainly for floodplains. However, some modern systems do provide river and coastal bathymetric data.

Raw LiDAR data are normally turned (often by the companies by which they are collected) into digital elevation models (DEMs), or ‘‘raster’’ grids, where elevation data are presented on a georeferenced square regular grid. Such grids have a cell size of 1m at best (in most UK studies), so that each cell elevation is based on many raw data point elevations averaged in some way. Data processing techniques include such tasks as feature extraction, filtering, and segmentation. For flood modelling applications, it is often necessary to perform a segmentation process, i.e., to separate the DEM into areas such as bare earth, vegetation, buildings, etc. These features may be extracted, resulting in a DTM, which contains only ground elevations [Mason et al 2003][Sithole and Vosselman 2004].

LiDAR-based DEMs and DTMs of the floodplains are used in different ways depending on the type of hydraulic model used. In 1D models they allow the extension of channel cross-sections onto the floodplains. In 1D+ models, they are used to calculate storage (volume vs. water level) curves for each FSC. In 2D- and 2D models they provide cell or node elevations [Bates et al 2000][Cobby 2003]. In all these models, water depths are calculated by subtracting DEM/DTM elevations from the computed water levels. The suitability of LiDAR for such applications has been extensively investigated [Gomes et al 1999], recently Neelz and Pender [Neelz and Pender 2006] reported on the impact of errors in LiDAR DEMs on the quality of 2D model predictions.

7.3.2 SRTM DEM

SRTM (Shuttle Radar Topography Mission) DEM data products result from a collaborative mission by the National Aeronautics and Space Administration (NASA), the National Imagery and Mapping Agency (NIMA), the German space

agency (DLR) and Italian space agency (ASI), to generate a near-global digital elevation model (DEM) of the Earth using radar interferometry [JPL2004]. 3 Arc Second (90 meter) DEMs for global coverage were developed from the SRTM C-band radar observations. The data was expressed in geographic coordinates (latitude/longitude) and was horizontally and vertically referenced to the WGS84 Geoid. The Shuttle Radar Topography Mission (SRTM) DEM of the study area was downloaded from the internet. SRTM DEM was used to compare with the created DEM for the flood plain of the study area.

7.4 Flooding Hazard Warning System

The topography of the river channel and the adjacent floodplain are able to be modelled by using terrain data which is discussed previously. The next step is focus on integrating the constructed river terrain with flood simulation model for flood hazard warning.

The proposed flooding hazard warning system is outlined in Figure 7.1, which consists of a number of input, output and visualisation elements.

The input elements contain digital GIS data, information on rainfall data, river water rising level monitoring information, and weather forecasting; the outputs include simulation results of predicted flood spreading, geometrical real 3D world reconstruction and visualisation, visualised flooding information for risk assessment and emergency response which will be discussed in future work; the core part of the system consists of terrain modelling using GIS data, aviation/satellite image segmentation and river object recognition and registrations, real time river surrounding regional geometric information analysis, and the flood model. In this method, we will focus on the details of reconstructing

the real world using digital GIS and satellite data, river object identification and real time flooding region growing.

The system depends heavily on flood simulation model, which is important for predicting inundation areas. For this reason, flood spreading model and the hydrodynamic model are proposed and implemented in the system which will be discussed in the following.

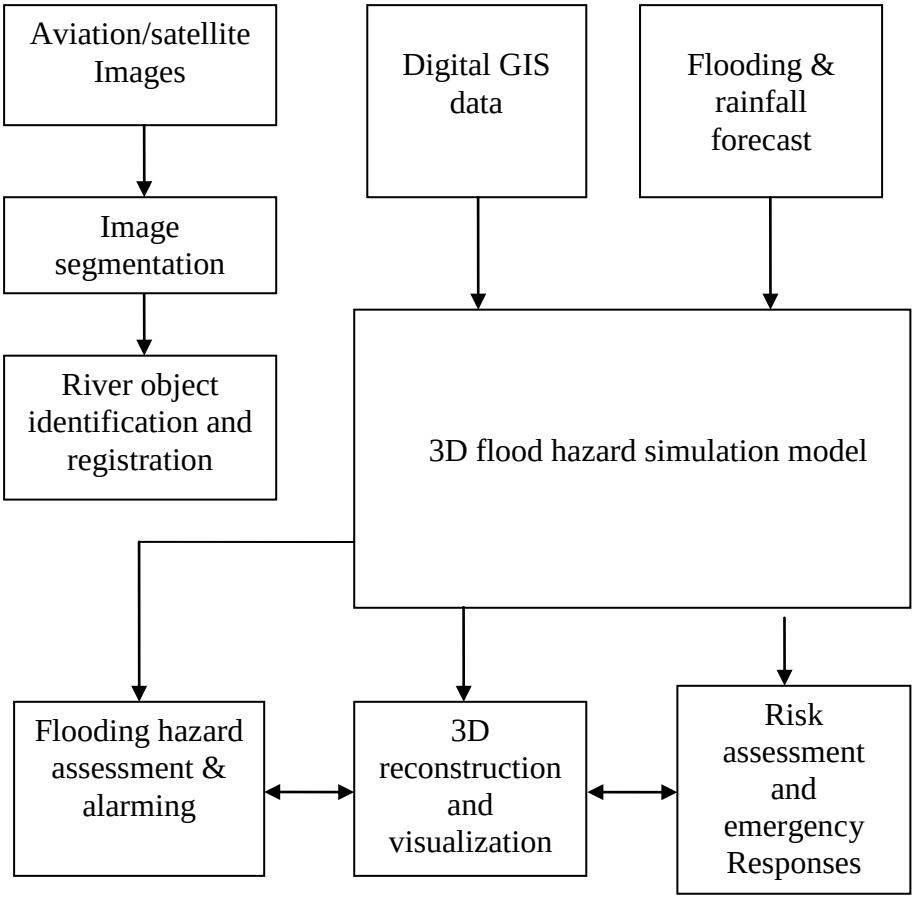


Figure 7.1 Flood hazard warning system

7.5 Flood Spreading Model

This model is developed for simulating the flood spreading process and obtaining the final inundation area in 3D environment. Standard gradient descent method for energy minimization is the key issue in the model. It uses a river

contour to approach the flooding area boundaries which are received by water level calculation. During the increase of water level, the original contour will move further due to gradient descent computation among terrain elevation data. This computation is accompanied by lots of dynamic contours which will be used as input for flood region search and merge process. Before we talk about important components of this flood spreading approach, some fundamentals of the model are introduced.

7.5.1 Fundamentals of Flood Spreading Model

In a complicated topography, the flood submersion can be divided into two main stages: (1) Flood spreading stage from the source of overflow to low-lying areas; (2) Final flood storage stage in low-lying areas. During the first stage, the flood spreading process is for searching region connectivity in low-lying areas. In the second stage, the process of formed flood storage is related to the volume of flood flow and drain capacity in that area. We define the difference between volumes of flood flow (F_i) and drain capacity (F_o) as flood storage capacity (F_s), that is:

$$F_{s=} F_i - F_o \tag{7.6}$$

Through looking for the lowest connected place in flood area, the start point of flood storage can be obtained. From this point, flood storage area expands. The more areas of flood storage it enlarges, the more flood volume it contains. By comparing flood storage capacity (F_s) with regional volume (V), if F_s is greater than V, that means the flood storage cannot contain this volume of flood flow, flood storage area should be enlarged and recalculated until F_s is equal to V, then the eventual flood storage area can be confirmed. Volumes of flood flow (F_i) can be gained from current water level calculation, hydrodynamic and hydrographical

models computation or result of prediction. Drain capacity (F_o) can be obtained from historical data or hydrodynamic and hydrographical models computation. Regional volume (V) could be calculated from elevation difference multiplies area of this region.

7.5.2 Flooding Image Spatial Segmentation

The first stage process is to identify the river objects via flooding image segmentation. To obtain the necessary information contained in the image of the flooding environment, the flooding image has to be segmented. An image spatial segmentation scheme is proposed in the current work by our active contours model. It will be then possible to calculate the river flow volume in real time for flood monitoring and forecasting once aviation or satellite images are available.

Figure 7.2 Original river image

Figure 7.3 A processed river image

Figure 7.3 shows the result of a processed river image, where the river object identified and river segments were registered with blue colour for further process. The original river image is shown in Figure 7.2.

After identifying the river object in satellite images, water level calculation is required to consider and carry out from terrain data.

7.5.3 Water Level Calculation Process

For modelling the flood region growing in the current system, a local earth region is applied to calculate river water level.

Algorithm for level of water calculation in the given profile is relatively complicated. Input is the flow rate but it is not possible to find out level only by putting into formula. On the other side, it is possible to count the flow rate *Q* with putting the level of water *h* and several other known values. That is why we change the level of water value *h* and count the flow rate *Q* in consecutive iteration in the algorithm. Iteration stops when value *Q* approximately matches the desired flow rate in the crossprofile. Then the actual level *h* represents flood level in the given profile.

For flow rate calculation we used the following formulas depending on the area, wetted perimeter, roughness of river bed and gradient of river bed:

$$Q = S \cdot v \tag{7.7}$$

$$v = c \cdot \sqrt{R \cdot i} \tag{7.8}$$

$$c = \frac{1}{n} \cdot R^{\frac{1}{6}} \tag{7.9}$$

$$R = \frac{S}{O} \tag{7.10}$$

After conversion $Q = S^{\frac{5}{3}} \cdot O^{\frac{-2}{3}} \cdot i^{\frac{1}{2}} \cdot n^{-1}$ (7.11)

where:

Q – flow rate *S* – flow area *O* – wetted perimeter

v – flow velocity *c* – velocity coefficient *R* – hydraulic radius

i – gradient *n* – roughness coefficient

The wetted perimeter O, the flow area S and (from that) the flow rate Q_{POC} is counted at each change of the water level. The values of roughness and gradient are known constants. If the water level exceeds any bank of the river bed, the flow calculation is divided into three parts (see Figure 7.4), for: the river bed, right bank and left bank. Each part has not only different values of area and perimeter but also roughness and gradient. Therefore we count the flow rate for each part separately and get the resultant flow for the given cross-profile by counting up the flow rates from separate parts. Algorithm counts the flow rates for water levels corresponding to heights of the left and right bank and maximum flow rate which we are able to count. The entered flow rate is tested for being less than the maximum one before calculation. If it is valid, the calculation continues.

Flow area:

S_1 – the river bed

S_2 – the left bank

S_3 – the right bank

Wetted perimeter:

O_1 – the river bed

O_2 – the left bank

O_3 – the right bank

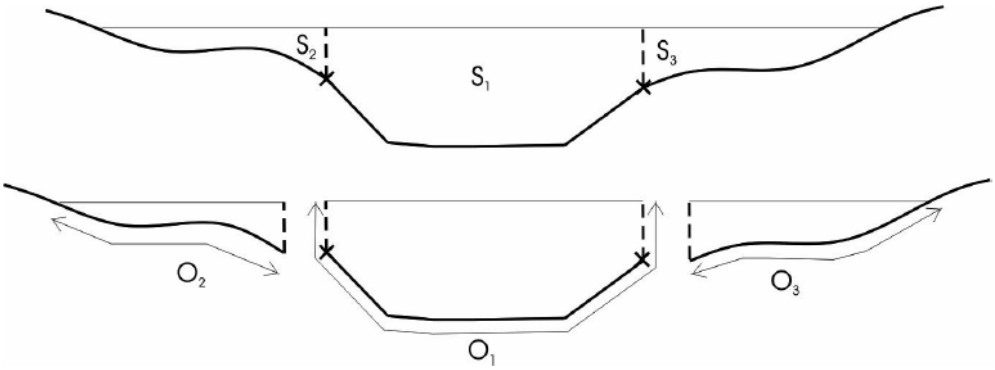


Figure 7.4 The method of cross-profile dividing into three parts

Height of:

H_L – the left bank line

H_R – the right bank line

Maximal flow rate:

Q_L – according to H_L

Q_R – according to H_R

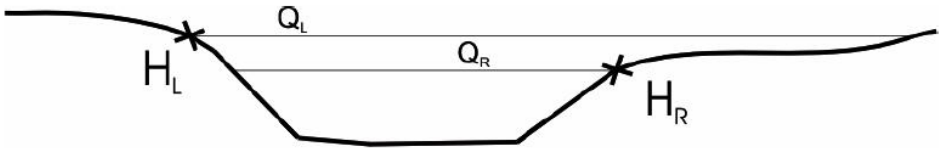


Figure 7.5 The flow rates according to the left or right bank line

Comparing the entered flow rate with flows Q_L and Q_R (the flows when the water level corresponds to the left or to the right river bank line – Figure 7.5) we find out whether the spill-over from the river bed happens and we get the initial level of water. There can arise four cases, see Table 1.1:

Condition	Schematic drawing	Description of situation	Initial level of water
$Q<Q_L$ and $Q<Q_R$		Level of water is not higher than right bank line. Calculation proceeds only in the river bed.	Lowest point in the river bed
$Q_R<Q<Q_L$		The water level overflows the right bank line. Calculation proceeds in the river bed and on the right bank.	Height of the right bank line
$Q_L<Q<Q_R$		The water level overflows the left bank line. Calculation proceeds in the river bed and on the left bank.	Height of the left bank line
$Q>Q_L$ and $Q>Q_R$		The water level overflows the right and also the left bank line. Calculation proceeds in the river bed and on both banks.	Height of higher of bank lines, max(Q_L, Q_R)

Table 1.1 Four cases of water spilling during the flood.

The water level rises in specific steps in iteration (initial step is set on 0.5 m). If we exceed a desired flow rate for the given level of water, we return one step back and reduce the step. Step after the first reduction has the value 0.05 m and during the next reduction is divided into halves. In the main it corresponds with interval bisection. The mentioned iteration is ended in the moment when the

calculated flow rate roughly equals the demanded flow in the cross-profile.

The algorithm of water level interpolation between individual cross-profiles firstly creates a new GRID. This GRID represents the flood model and has the same cell size and spatial location as DTM. Furthermore, we interpolate the value for each cell of a new GRID. Interpolation is linear and by perpendiculars: At first we find the crossprofiles between which the interpolated point is placed. We create the perpendiculars from point to both crossprofiles and count the distance between the point and each of the profiles (d_1, d_2). Ratio of distances is used as an interpolation ratio. With this value we multiply the difference of flood level in neighbouring profiles. Afterwards we add this result to that one of the neighbouring cross-profiles which has lower elevation above sea level. This final value represents the flood elevation in the given place which will be put into the GRID cell. The interpolation formula:

$$h = h_1 + \frac{d_1}{d_1 + d_2} \cdot (h_2 - h_1) \tag{7.12}$$

7.5.4 Standard Gradient Descent Method for Energy Minimization

To get a real flood model, we further have to crop the resulting grid with DTM. While simulating flood in complicated landform, the connectivity between different regions has an important effect on flood spreading direction. In this stage, due to gravity, flood flows into low-lying area. The flow direction may be changed when it is subject to influence of high or low elevation. Flood flowing process can be treated as search for optimization problem, so it requires a specific algorithm to achieve this goal.

We use the standard gradient descent (or steepest descent) method integrated with our digital terrain model to minimize the energy function. This could

determine the flood flow direction and inundatiuon area. The energy function guides the motion of the flood contour toward the final inundation boundaries.

$$T(C, f_1, f_2) = \varepsilon(C, f_1, f_2) + \mu P(C) + vQ(C) \quad (7.13)$$

$$\varepsilon(C, f_1, f_2) = \int_{\Omega} \varepsilon_x(C, f_1(x), f_2(x))dx \quad (7.14)$$

$$P(C) = \int \frac{1}{2}(|\nabla C(x)| - 1)^2 dx \quad (7.15)$$

$$Q(C) = \int_{\Omega} \delta(C(x))|\nabla C(x)|\, dx \quad (7.16)$$

Where μ and v are nonnegative constants. We define functional $\varepsilon(C, f_1, f_2)$ to minimize the integral of ε_x over all the centre points x in the image domain Ω. The functions $f_1(x)$ and $f_2(x)$ are weighted averages of the intensities in a neighbourhood of x which fit image intensities near the point x.

Keeping f_1 and f_2 fixed, we minimize the energy functional $T(C, f_1, f_2)$ with respect to C using the standard gradient descent method by solving the gradient flow equation as follows:

$$\frac{\partial C}{\partial t} = -\delta_\omega(C)(\lambda_1 m_1 - \lambda_2 m_2) + v\delta_\omega(C) div\left(\frac{\nabla C}{|\nabla C|}\right) + \mu(\nabla^2 C - div(\frac{\nabla C}{|\nabla C|})) \quad (7.17)$$

The smoothed Dirac delta function is δ_ω , m_1 and m_2 are the functions:

$$m_i(x) = \int L_\sigma (y - x)|I(x) - f_i(y)|^2 dy\,, i = 1,2 \quad (7.18)$$

The above (7.17) is the level set evolution equation to be solved in the proposed method. The $-\delta_\omega(C)(\lambda_1 m_1 - \lambda_2 m_2)$ term is derived from the data fitting energy, and, therefore, is referred to as the data fitting term. This term plays a key role in the proposed model, since it is responsible for driving the flood contour toward final inundation boundaries. The second term $v\delta_\omega(C)\mathrm{div}\left(\frac{\nabla C}{|\nabla C|}\right)$ has a length shortening or smoothing effect on the zero level contours, which is

necessary to maintain the regularity of the contour. This term is called the arc length term. The third term $\mu(\nabla^2 C - \mathrm{div}(\frac{\nabla C}{|\nabla C|}))$ is called a level set regularization term, since it serves to maintain the regularity of the level set function.

An example has been listed to test the standard gradient descent method. The initial flood level has been calculated by the water level calculation process. Figure 7.6 shows the initial stage of flood contour. Then the contour will extend to capture the potential flood inundation boundaries by comparing the internal and external energy. Figure 7.7 shows the different stages of contour spreading process. When the internal energy is equal to external energy, the whole process is finished. The final inundation boundaries have been obtained at t=250s.

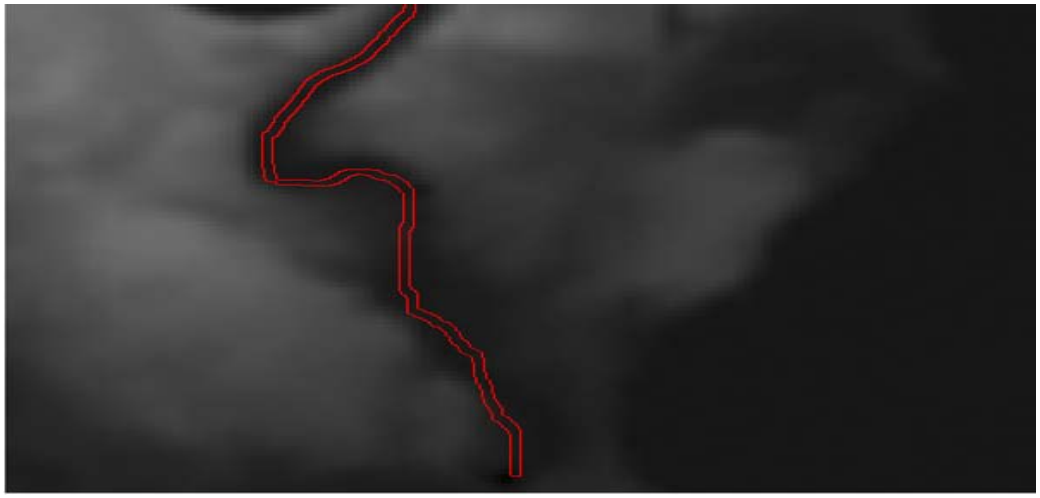
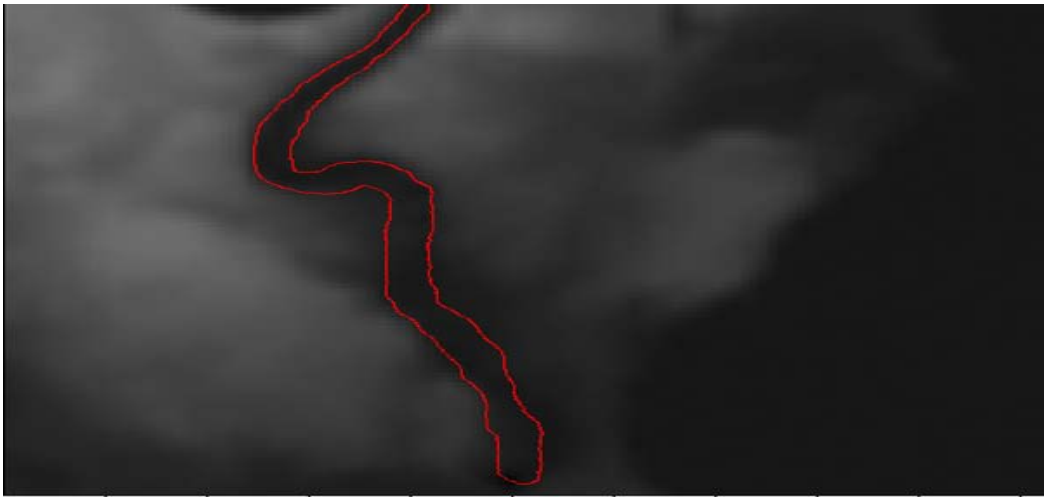
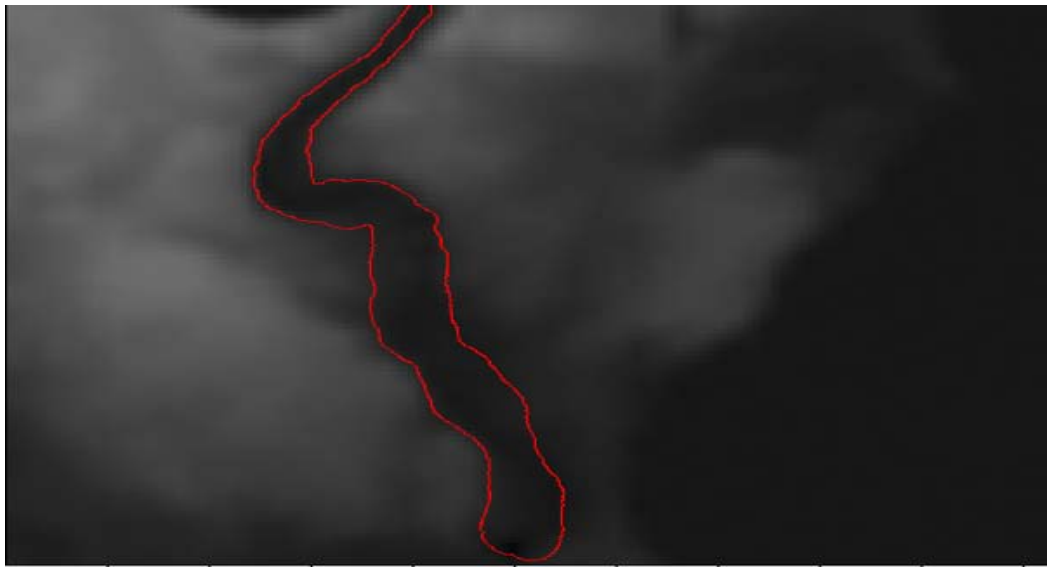


Figure 7.6 Initial stage of flood contour

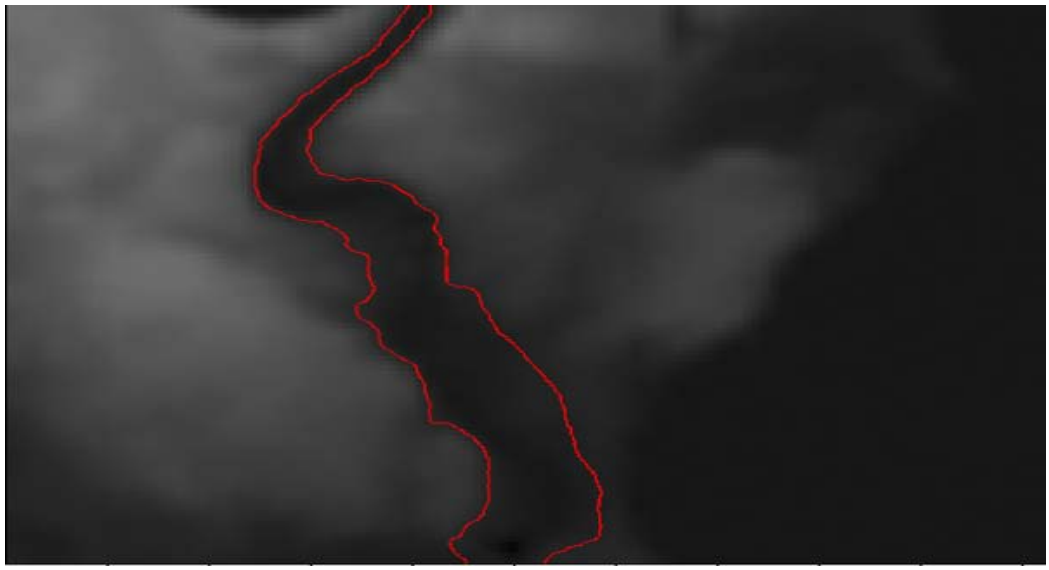
t=51s



t=106.3s



t=210s



t=250s

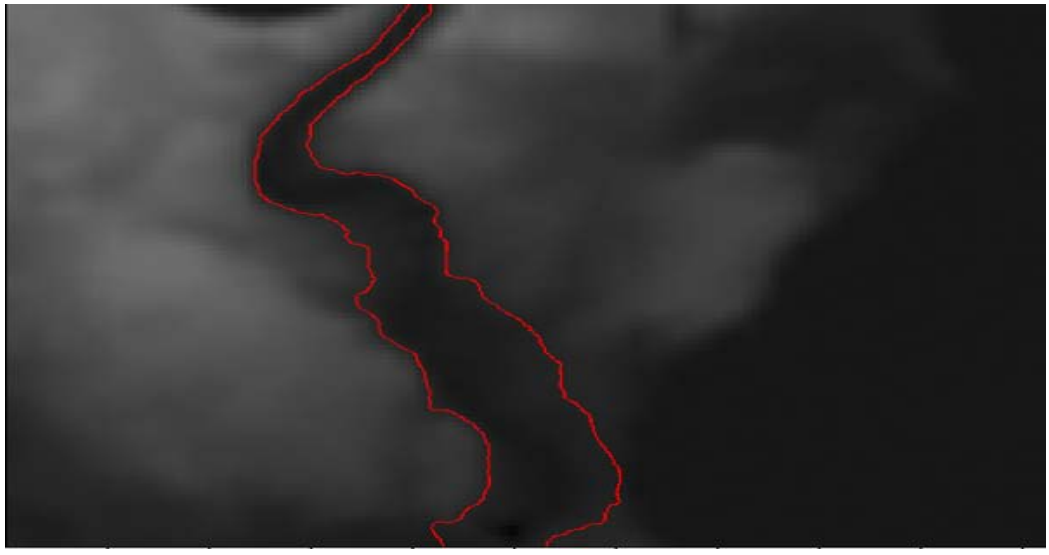


Figure 7.7 Results with SRTM data after 51s, 106.3s, 210s and 250s.

7.5.5 Flood Region Search and Merge Process

The flood spreading contours are obtained through standard gradient descent method; those contours will be used as input for predicting inundation areas. In order to achieve this goal, the flood region search and merge process is proposed and implemented. The flood region search and merge process is conducted by a weighted cell region adjacency graph. The similarity measure associated with two neighbouring cells is found by firstly calculating the mean intensity values of all vertices associated for each cell, and then for the boundary between the two. The down cell will be flooded if and only if the flooded cell's (up cell) mean value is greater than the down cell's mean value, and greater than the minimum value of the common boundary. A connected cell is identified as part of a flooded cell of the flood region if its elevation level is under a dam threshold of a neighbouring flooded cell. The process will be conducted in all eight neighbouring directions around the cell assessed. Flooded cells will be merged to become a large cell, and so on. The configured flood region consists of all flooded cells connected together and will then be used as a flooding field. It is obvious that a flood that has occurred may consist of more than one flooding field. Because the cell is used for constructing the local earth region model and for its high efficiency, a river cell element scheme is implemented in the process.

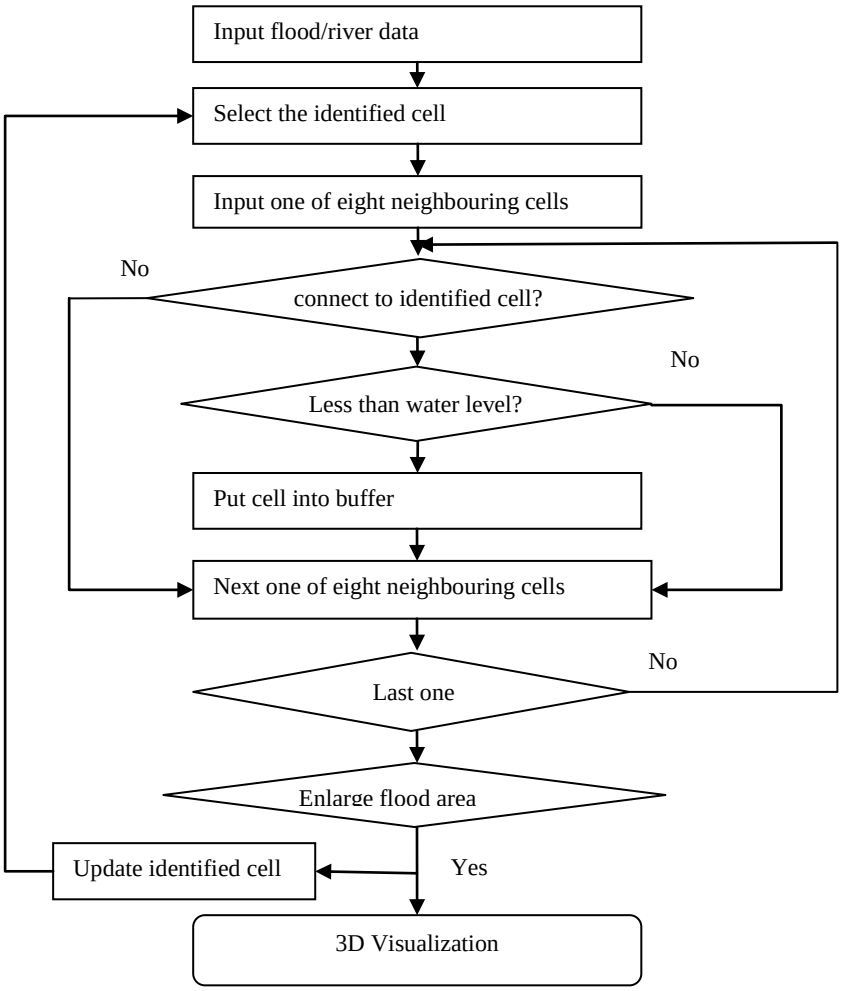
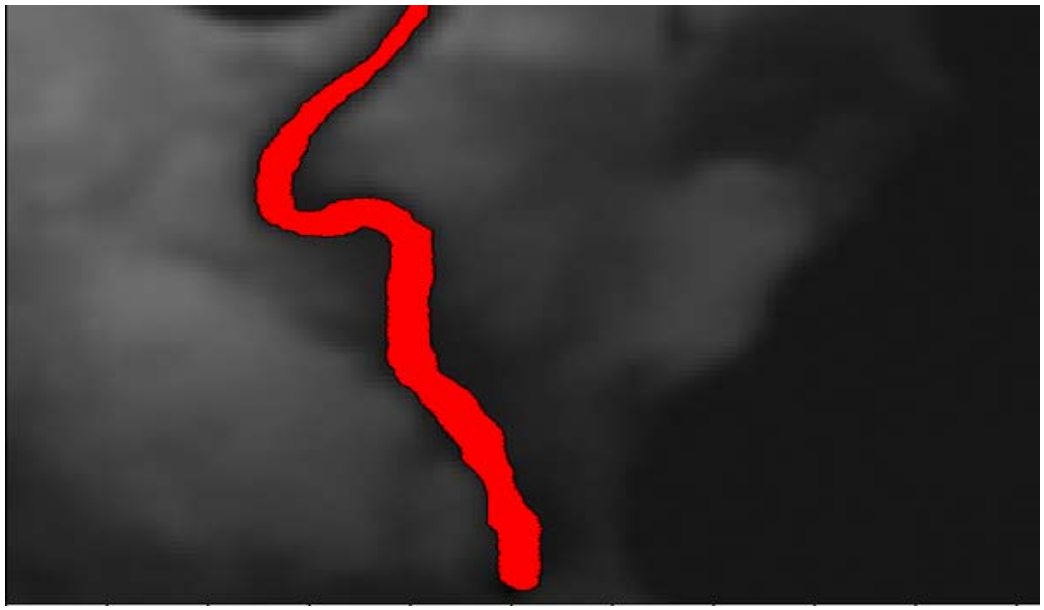


Figure 7.8 Flood Region Search and Merge Process

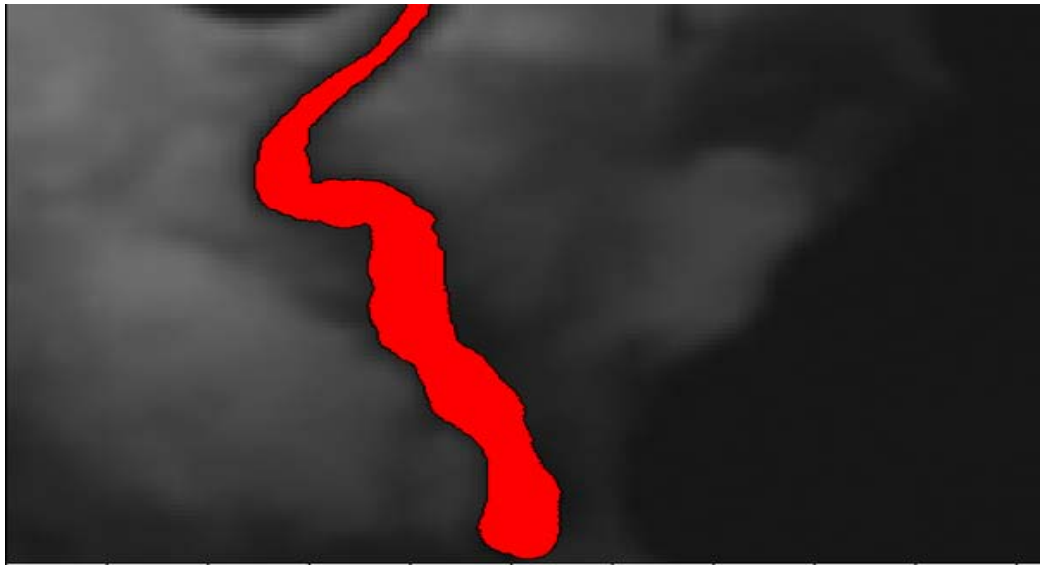
The flood will start at a cell or a location at a river dam of the region, with greater danger for flooding occurring due to its geometric profile. It will then spread fast over the entire potential flood region. The process can be described as shown in the above flowchart in Figure 7.8.

An example has been conducted to test the flood search and merge process method. The path of flood flow is to low-lying area. Figure 7.9 presents different flood inundation stage with red colour.

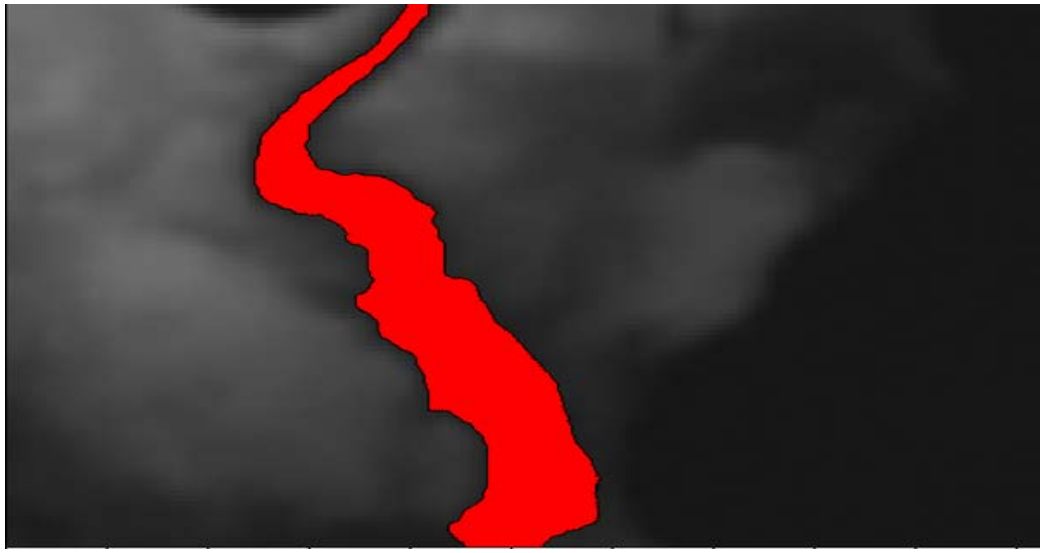
t=51s



t=106.3s



t=210s



t=250s

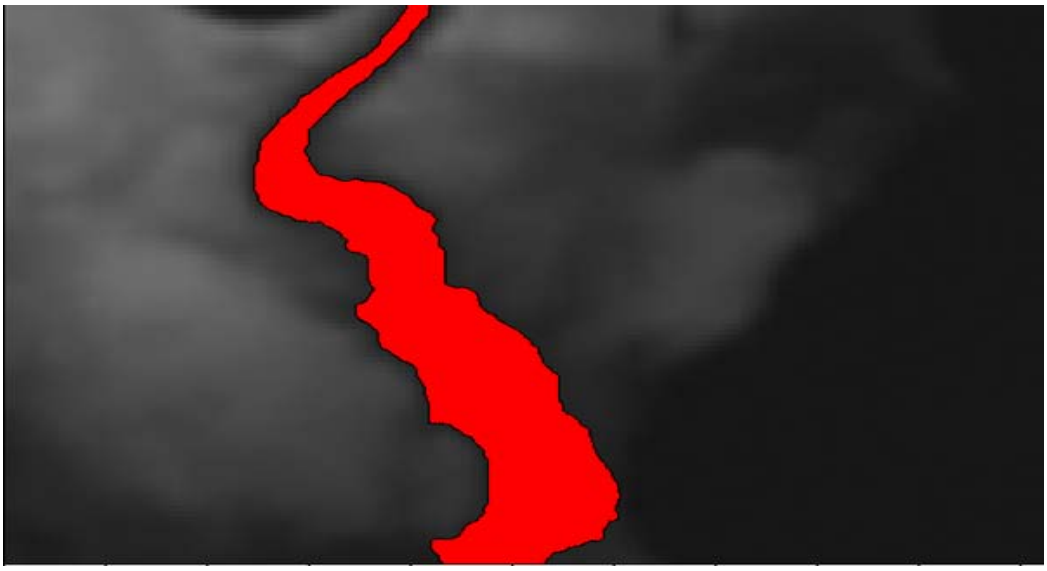


Figure 7.9 Calculated Flood Area by Flood Search and Merge Process

7.6 The Finite Volume Hydrodynamic Model

For accurate flood simulation, numerical models based on two-dimensional shallow water equations are commonly used. In small-scale dense area, we use a finite-volume method for solving the shallow water equations [Zoppou and Roberts 1999]. The study area is represented by a mesh of triangular cells in which water depth *h*, and horizontal momentum (*uh*, *vh*), are determined. The size of the triangles may be varied within the mesh to allow greater resolution in regions of particular interest.

Finite volume methods have several advantages over finite difference and finite element approaches. Finite volume methods combine the simplicity of finite difference methods with the geometric flexibility of finite element methods. FVM can be considered as finite difference methods applied to the differential conservative form of the conservation laws in arbitrary co-ordinates. Thus the methods can be applied using an unstructured grid system as FEM; generally FVM need less computational effort than FEM.

FVM are based on the integral form of the conservation equations, thus a scheme in conservation form can easily be constructed to capture shock waves (shock-capturing property). By discretization of the integral form of the conservation equations, the mass and momentum remain conserved [Hirsch 1990]. The key problem in FVM is to estimate the normal flux through each cell interface. There are several algorithms to estimate this flux. The set of shallow water equations is hyperbolic and therefore it has an inherent directional property of wave propagation.

The numerical solution of the shallow water equations for flood wave propagation over real domains poses three specific problems. The first problem is the simulation of the fronts or abrupt water waves that can be represented numerically as a propagating discontinuity. This problem can be considered as solved since the late eighties or early nineties. The second problem derives from abrupt changes in bathymetry. As long as the bottom surface remains sufficiently smooth, most numerical techniques provide an accurate solution of the flow, but if the bottom surface is very rough the majority of methods fail. The last problem especially arises when these schemes are applied to study the wave front propagation over dry bed.

The essential aspects of the method used are illustrated in the following:

a). Governing equations

The mathematical model used is based on the two-dimensional shallow water equations. This model is obtained by averaging the Navier-Stokes equations along vertical direction under some assumptions. The shallow water wave equations are a system of differential conservation equations of the form:

$$\frac{\partial U}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial G}{\partial y} = S \tag{7.19}$$

where $U = [h\ uh\ vh]^T$ is the vector of conserved quantities; water depth *h*, *x* momentum uh and *y* momentum *vh*. Other quantities entering the system are bed elevation *z* and stage (absolute water level) *w*, where the relation $w = z + h$ holds true at all times. The fluxes in the *x* and *y* directions, *E* and *G* are given by

$$E = \begin{bmatrix} uh \\ u^2h + gh^2/2 \\ uvh \end{bmatrix} \quad \text{and} \quad G = \begin{bmatrix} vh \\ vuh \\ v^2h + gh^2/2 \end{bmatrix} \tag{7.20}$$

and the source term (which includes gravity and friction) is given by

$$S = \begin{bmatrix} 0 \\ gh(S_{0x} - S_{fx}) \\ gh(S_{0y} - S_{fy}) \end{bmatrix} \tag{7.21}$$

where S_0 is the bed slope and S_f is the bed friction. The friction term is modelled using Manning's resistance law

$$S_{fx} = \frac{u\eta^2\sqrt{u^2+v^2}}{h^{4/3}} \quad \text{and} \quad S_{fy} = \frac{v\eta^2\sqrt{u^2+v^2}}{h^{4/3}} \tag{7.22}$$

in which η is the Manning resistance coefficient.

b) Finite Volume Method

The two dimensional physical domain is decomposed into elementary quadrilaterals; each of them has to be considered as an elementary control volume. The four straight sides enclosing it form its boundary.

Figure 7.10 shows a single elementary control volume that is surrounded by four walls, whose length is ds_r (r=1, 4). The outward normal unit vector to each wall is also displayed, under the symbol *n*.

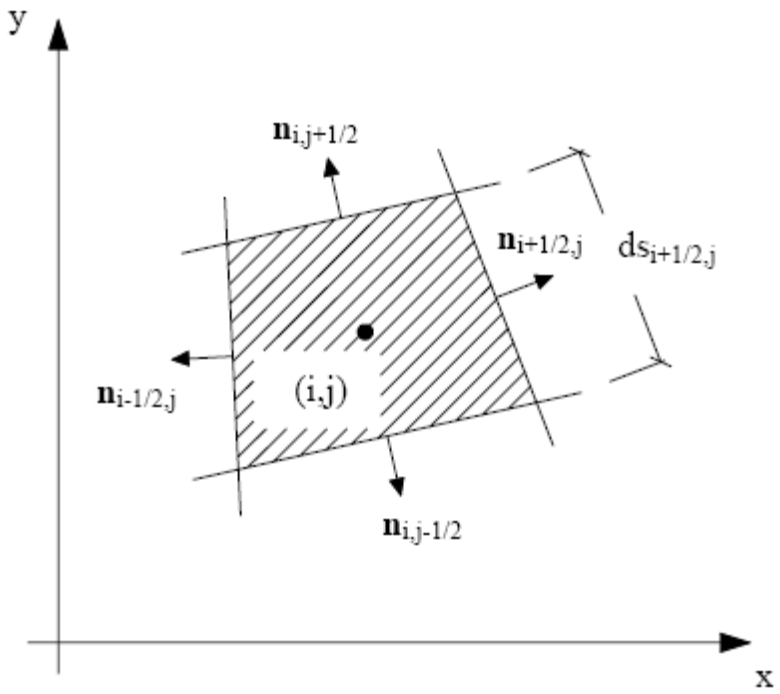


Figure 7.10 Generic control volume end notation

Integrating equation 7.19 over an arbitrary control volume *V* and calling U_{ij} the average of *U* over cell, one obtains:

$$\frac{\Delta U_{ij}}{\Delta t}\Delta V = -\oint_1^S (F \cdot n)dS + \int_1^v SdV \tag{7.23}$$

c) Source terms

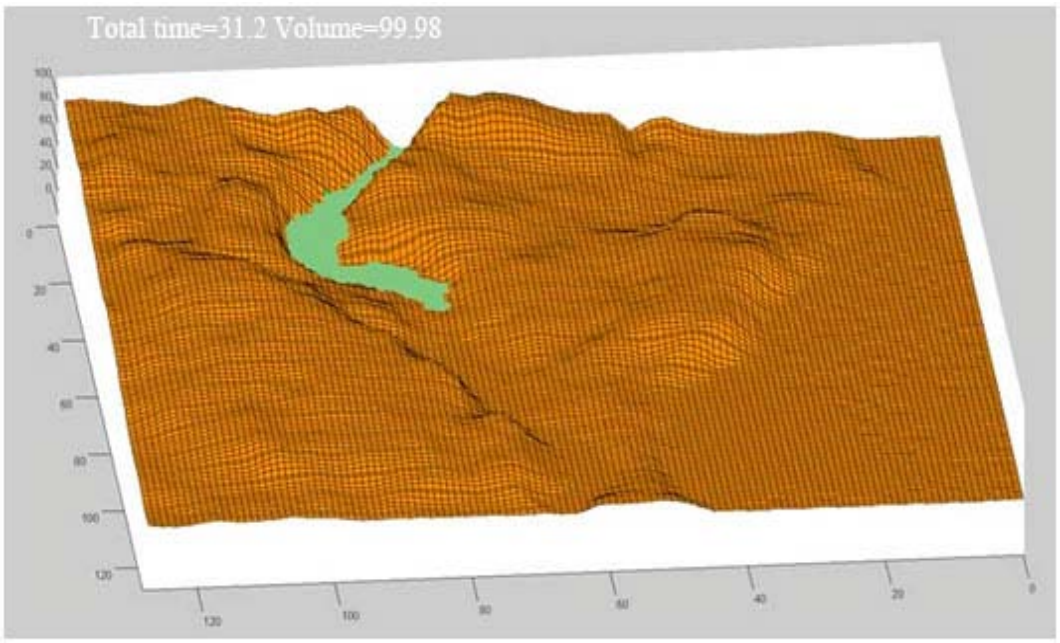
The numerical treatment of source terms has a great influence on the accuracy of the scheme. In this model the source term vector is firstly split into its two parts, respectively dependent from the bottom slope and from the bed friction. The first part is treated in an explicit way, while the second part is treated in a semi-implicit way.

$$S = S_0 + S_f \tag{7.24}$$

where

$$S_0 = \begin{bmatrix} 0 \\ ghS_{0x} \\ ghS_{0y} \end{bmatrix};\ S_f = \begin{bmatrix} 0 \\ -ghS_{fx} \\ -ghS_{fy} \end{bmatrix} \tag{7.25}$$

The following is an example of simulating one part of river flood by using our finite volume hydrodynamic model. The aim of this test case is the study of the behaviour of the model on dry bed, with particular attention to the front wave motion. The computational domain is discretized into square cells with Δx = 0.5 m, Δy = 0.5 m. The Manning coefficient is 0.02 s/m$^{1/3}$. The outflow boundary in the left end is set to be an open boundary. The Figure 7.11 shows the computed results of flood flow in the river Wye at different time. At t=31.2s, flood wave reaches the embankment of the river Wye. The water surface outside the embankment ascends due to the influence of the embankment and overtopping flow occurs. Subsequently, more and more water flows into the river, while flood wave continues to propagate along the stream channel. At t=100s, t=137.28s and t=652.48s, flood waves reach different locations and show that the numerical model gives a realistic prediction of the inundation area. The following figure also shows that the water surface changes due to the deflection and reflection of irregular boundaries are well reproduced by the numerical model.



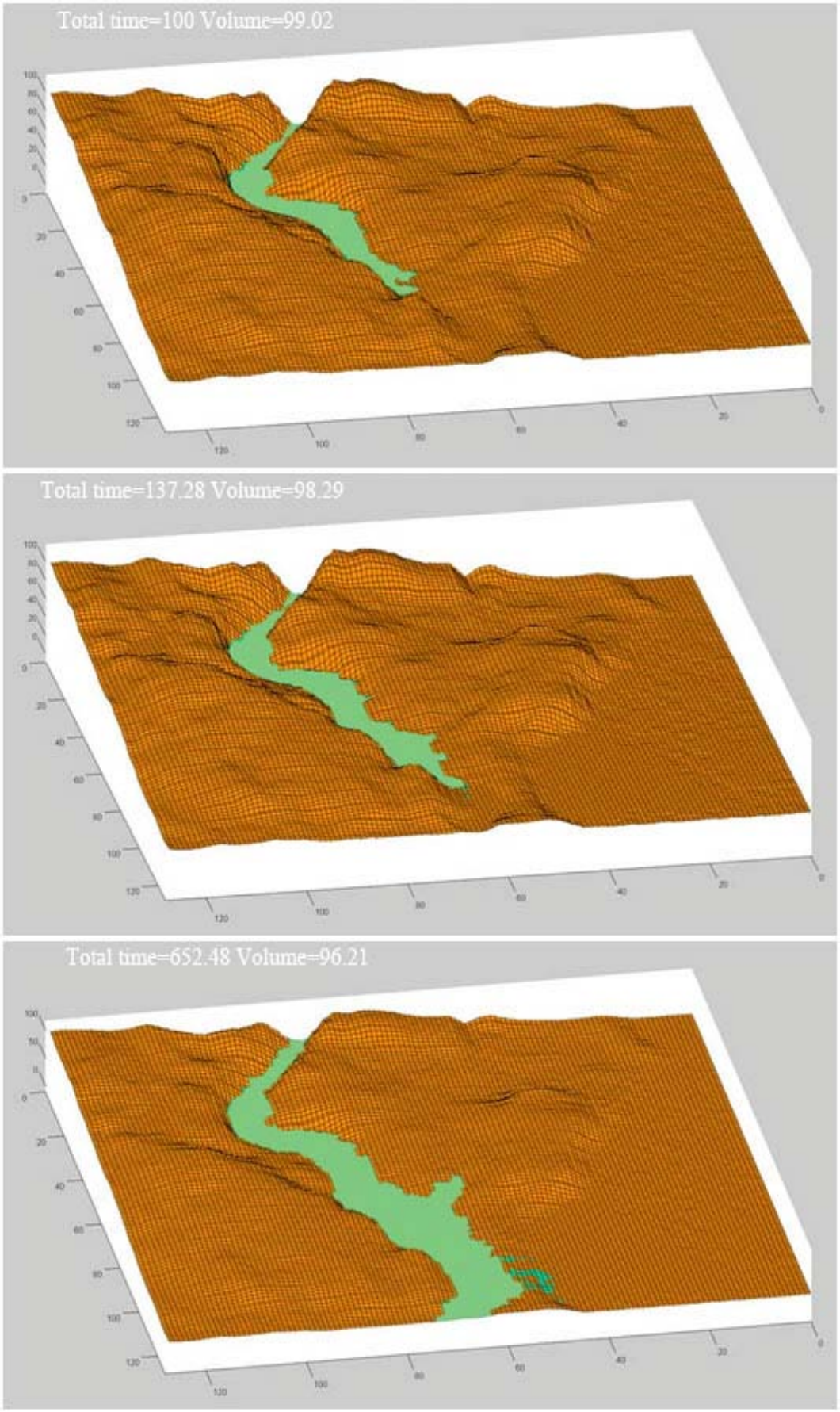


Figure 7.11 Commutated Flood Area with DEM by shallow water equations

7.7 Conclusion

This chapter proposes two novel flood simulation models which are capable of dealing with large scale and urban area detailed flooding situation. For large scale flooding, flood spreading model is proposed which is made up of four parts: (1) Flooding image spatial segmentation (2) Water level calculation process. (3) Standard gradient descent method for energy minimization. (4) Flood region search and merge process. For dense urban area, the finite volume hydrodynamic model is adopted which can provide much more accurate prediction. Some experiment results show our method can handle dynamic flood behaviour with fast speed and great accuracy in real time.

Chapter 8 Flood Simulation Case Studies

8.1 Introduction

Our flood simulation models have been developed to simulate flood inundation due to the conditions of the water level of a river overrunning the height of embankments or dams. The model adopts the raster grid so that GIS DEM can be directly imported into the terrain model. By using flood spreading model, finite volume hydrodynamic model and manipulating GIS data, it is able to predict the inundation area effectively.

In this chapter, first flood simulation test has been conducted for validating the performances of both flood spreading model and finite volume hydrodynamic model. The other four flood case studies have been simulated for evaluating the performance of the flood spreading model only. This is because the finite volume hydrodynamic model is a complement to the flood spreading model.

The studies achieved two main objectives: implementing a flood spreading process within 3D environment for predicting potential inundation area precisely; providing useful information to make a right decision when a real flood occurs.

The case studies are listed in the following:

- Case 1 is for simulating tidal flooding at Chepstow in Wales, which tests the urban flood situation. The 3D environment of Chepstow is modelled and reconstructed separately. Flood spreading process with inundation extent map at different time is shown with an easy to use interface.
- Case 2 is selected to test the capability of this simulation model to deal with real life problems with complicated topography. The study area has been chosen at Stejaru in Romania. The simulated flood area is very close

to the real inundation area which validates that our model is robust and computationally efficient.

- Case 3 is for simulating a potential flood of the London and Thames Estuary at UK. The simulated water levels along the River Thames show consistency with the real flood situation which gives better results than a 2D modelling approach.
- Case 4 is for simulating the flood affecting the city of Bewdley at UK. The flood spreading process is shown within different time and the flood inundation area is presented for further disaster assessment.
- Case 5 is for simulating a potential flood of a region of Tewkesbury in England. The flood spreading process is shown during different time and the small scale detailed inundation area is presented for further disaster evaluation.

8.2 Case Study of Chepstow Flood

Chepstow is a historic town located at the lowest established bridging point on the River Wye, some 5km upstream of its confluence with the River Severn. Much of Chepstow has been built upon sharply rising ground but the west of the Wye is a relatively flat low-lying area. This area was formerly associated with the town's history as a port and ship-building centre. A variety of old residential and commercial properties in that part of the town are subject from time to time to tidal flooding.

8.2.1 Description of Study Area

In recent years, extreme high tides have caused flooding on average once every five years. The December 1981 river level equalled the previously highest

recorded event in 1883 of 9.00 metres above Ordnance Datum and a total of 32 properties were flooded. Significant flooding also took place in 1989, 1990, 1994, and 1997 and there was localised flooding at Christmas 1999 [IFPFC]. The study area of Chepstow is shown in Figure 8.1 which is attained from Google Map.

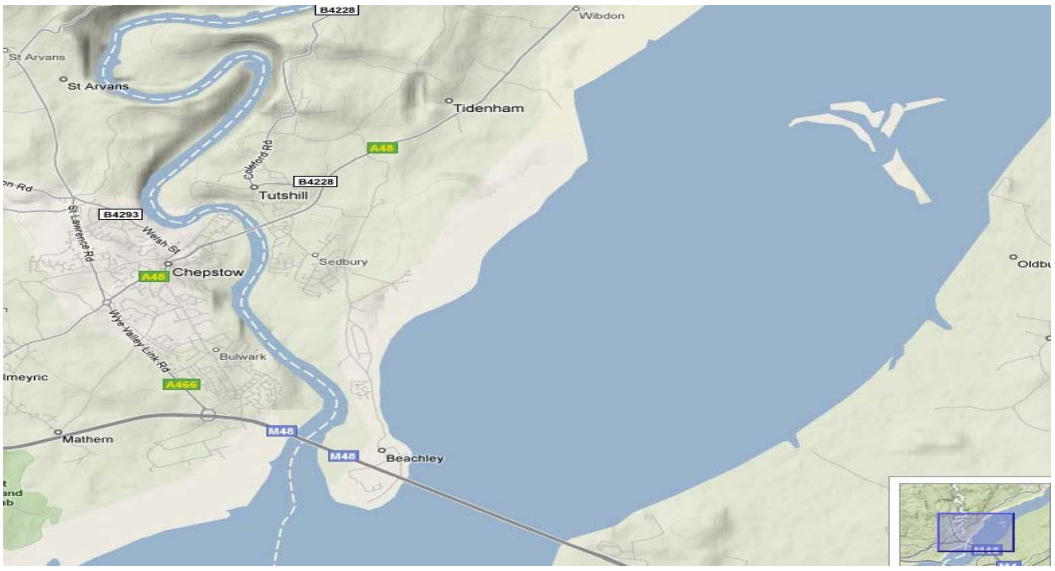


Figure 8.1 Location map of the River Wye and the study area (from Google map)

8.2.2 Available Data

This real city area is the subject of the simulation study. The case of flooding is in the upper part of the River Wye. The grid for the analyzed area was generated by using 3D real world modelling. The texture map for the surface of Chepstow terrain is obtained from Google Earth. The analyzed domain is 1000m x 500m [EDOS]. The forcing condition is a discharge of 1100 m^3/s, which applied suddenly by a flood wave that moves down the centre of the city. The time step is 10 seconds and the total duration of simulation is 800 seconds.

8.2.3 Texture Map Analysis and Segmentation

The texture map of Chepstow is obtained from Google Earth in Figure 8.2. Before modelling and reconstructing this 3D Chepstow terrain, image analysis such as specific areas segmentation is important.

Figure 8.2 Texture map of Chepstow

It is important to identify the river area in the texture map. Figure 8.3 represents the initial snakes generated from vector-valued scanning algorithm. Figure 8.4 shows the successfully recognised river with blue colour.

Figure 8.3 Initial active contour of river

Figure 8.4 Segmented river with blue colour

For sea area, we first assign initial contours (Figure 8.5). Then All of these initial snakes will move toward the river boundary (Figure 8.6). Figure 8.7 and Figure 8.8 show the final optimized contour and identified area with cyan colour after several iterations.

Figure 8.5 Initial active contours of sea

Figure 8.6 The contours after several iterations

Figure 8.7 Final contours of sea area.

Figure 8.8 Segmented sea area with Cyan colour

In the following part, Multi-seed region growing method is used for identifying the forest area. Figure 8.9 shows the initial segmented forest area. Figure 8.10 represents the whole segmented forest area with green colour. Figure 8.11 represents the initial snakes generated from vector-valued scanning algorithm. The final contours of urban area are successfully extracted which are shown in Figure 8.12. Figure 8.13 shows the segmented urban area with yellow colour.

Figure 8.9 Initial segmented forest area

Figure 8.10 Segmented forest area with green

Figure 8.11 Initial active contours of urban area

Figure 8.12 Final contours of urban area

Figure 8.14 shows the processed image of Chepstow integrated with previous recognised river, sea, forest and urban areas.

Figure 8.13 Segmented urban area

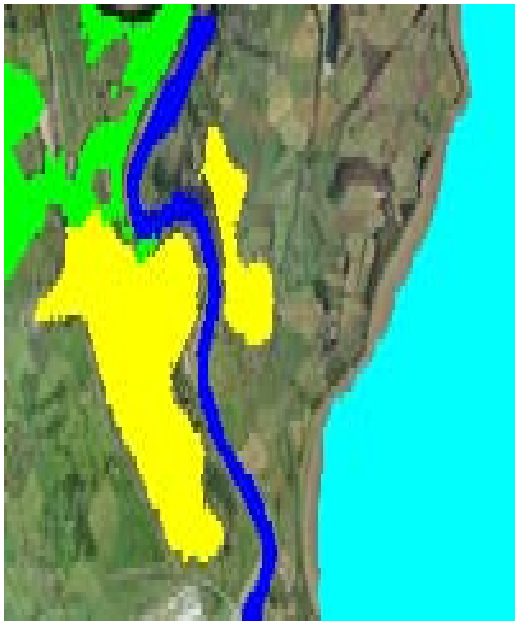


Figure 8.14 Segmented image of Chepstow

8.2.4 Digital Terrain Modelling and Reconstruction

For modelling the 3D environment of Chepstow, we use the proposed method in chapter 3. SRTM data is used for generating 3D geometry of Chepstow and the satellite image of Chepstow is obtained from Google Earth. The generated terrain geometry is overlaid by satellite image corresponding to the same geographic area which is presented in Figure 8.15. For terrain reconstruction, at first, we need to segment the image and select the objective areas for further study.

Figure 8.14 shows the segmented river, sea, forest and urban areas by using our active contours method and region growing approach which is mentioned in chapter 4 and chapter 5.

Figure 8.15 3D environment of Chepstow

Then simple house model and forest model are used to reconstruct the 3D environment of Chepstow by intelligent reconstruction method mentioned in chapter 5. Figure 8.16 and Figure 8.17 show the close distance view of reconstructed urban and forest area of Chepstow.

Figure 8.16 The close distance view of the Chepstow region

Figure 8.17 The close distance view of forest area

Figure 8.18 shows the complete reconstructed environment of Chepstow which is a good platform for doing flood simulation.

Figure 8.18 3D reconstructed environment of Chepstow

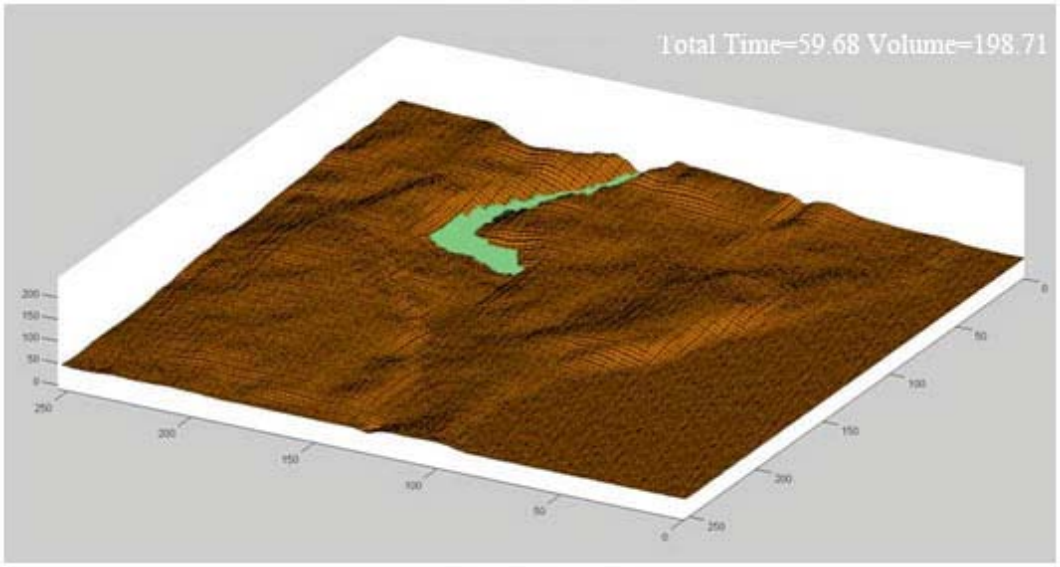
8.2.5 Simulation Results

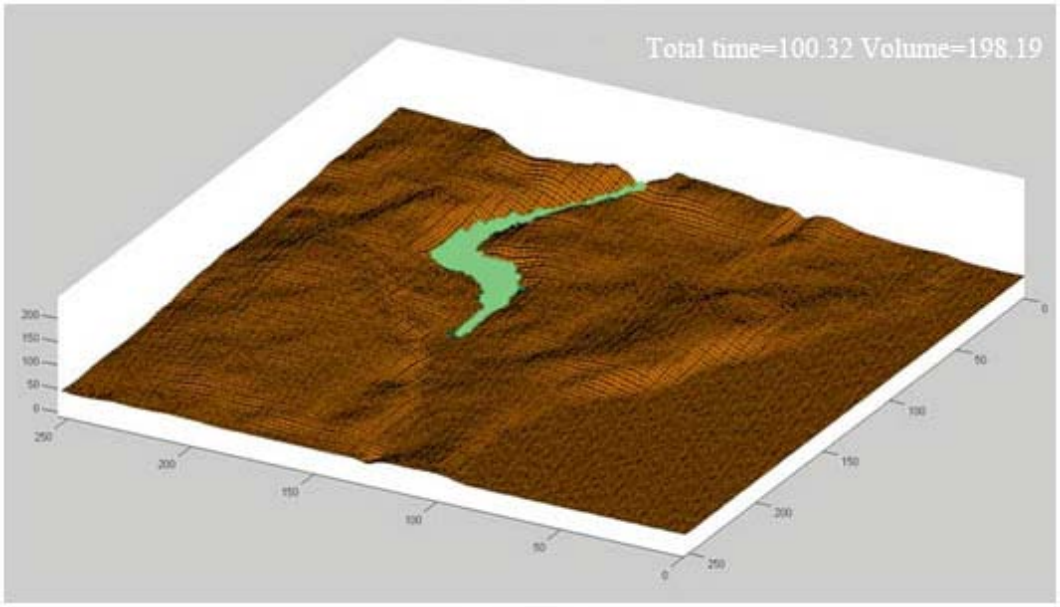
The test has been conducted for simulating the potential flood of the city Chepstow in Wales. This is for small scale flood simulation which contains more detailed information. The grid for the analyzed area was generated by using our 3D real world model. The analyzed domain is 1000m x 500m. The River Wye that flows through centre of the city is used for testing. An easy to use interface for this flood simulation is designed and with several functions presented in Figure 8.19.

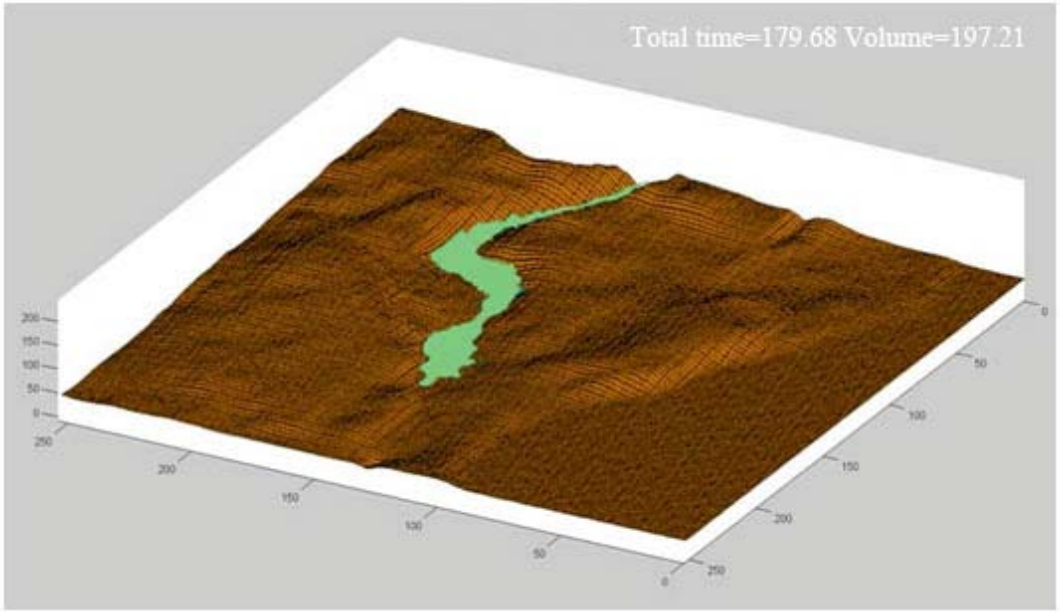
Figure 8.19 Interface of flood simulation system.

8.2.5.1 Results from Finite Volume Hydrodynamic Model

In Figure 8.20, we have computed dense Chepstow flood flow at different stages by using the finite volume hydrodynamic model. This numerical model based on two-dimensional shallow water equations can predict the flood inundation areas precisely.


Total Time=59.68 Volume=198.71


Total time=100.32 Volume=198.19


Total time=179.68 Volume=197.21

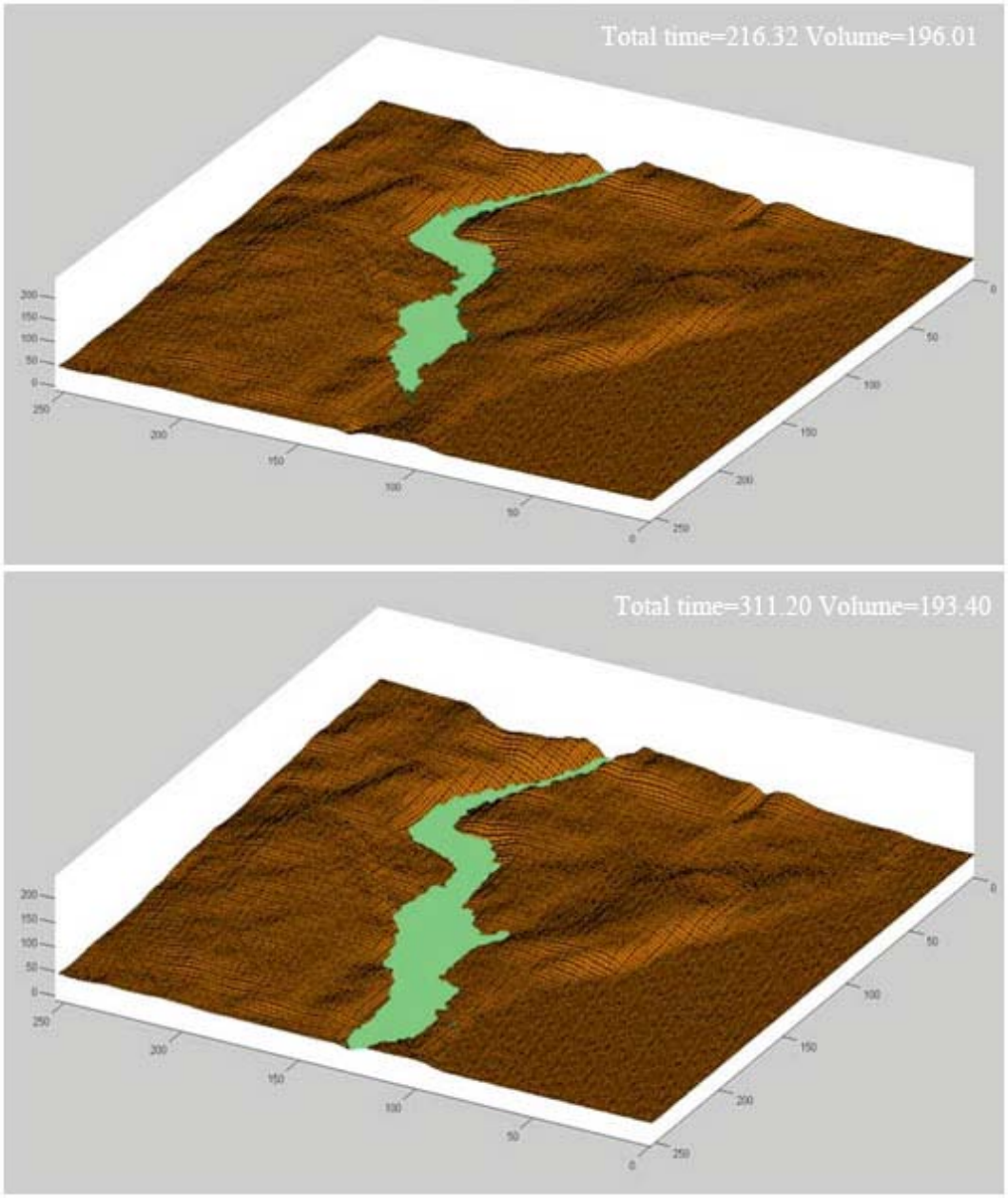


Figure 8.20 Computed Dense Chepstow flood flow at different stages

8.2.5.2 Results from Flood Spreading Model

When the river water starts overflowing its bank, flood hazard happens. The computed water elevation surfaces at t=50s, t=200s, t=400s, t=600s and t=800s are presented in Figure 8.21 which is based on flood spreading model.

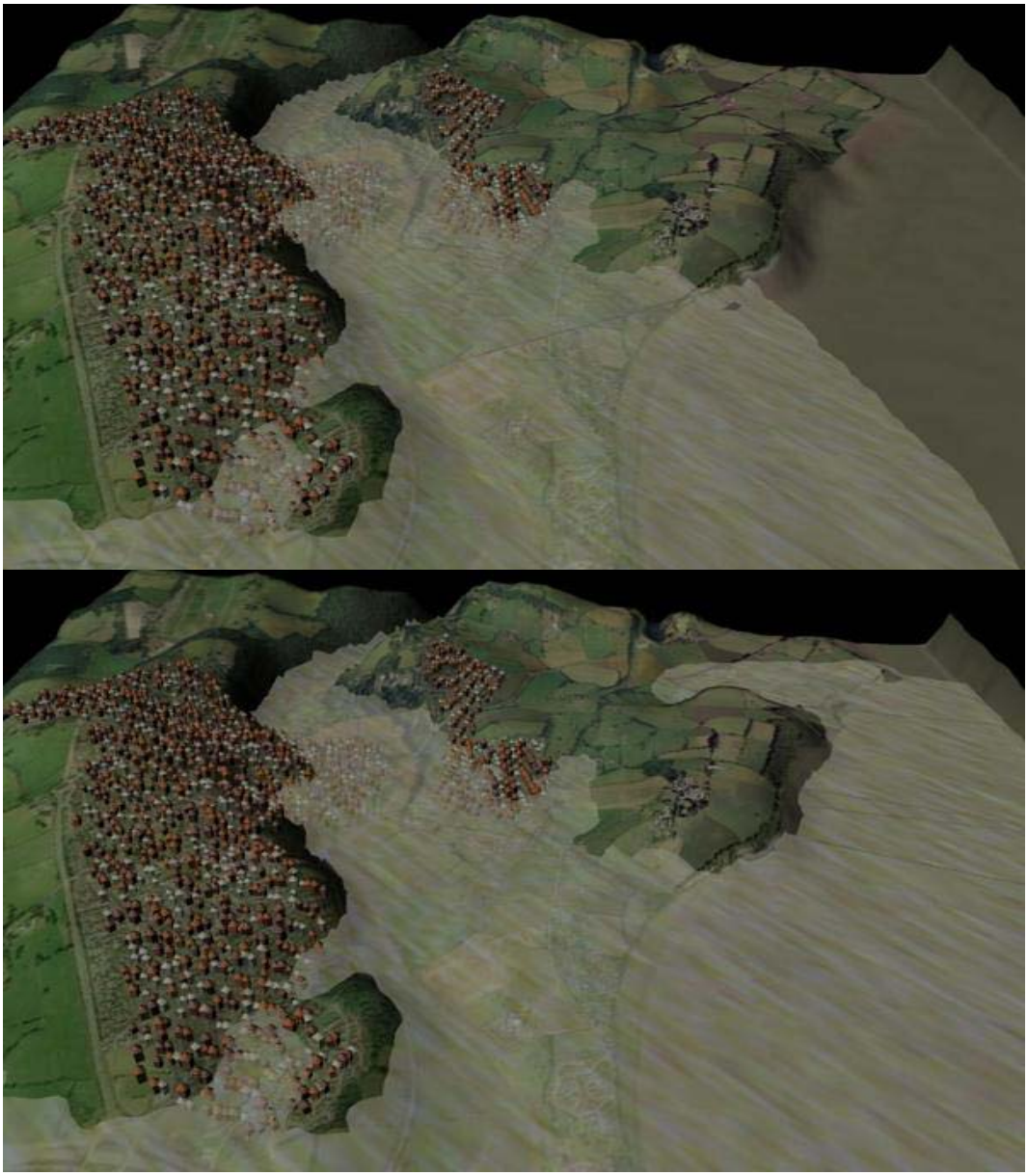


Figure 8.21 Chepstow flood spreading at different stages

8.2.6 Discussion

In this flood simulation test, satisfied experiment result has been achieved. At the same time, some other aspects of this work needs to be reconsidered. (1) In the event that we can add the damping coefficient and infiltration influence of different types of soil in the flood spreading model construction, it can increase the quality and prediction precision of simulation process. (2) Flood regional volume V

calculation depends on precision of elevation grids, the more dense grids we get, the more accurate prediction of inundation area we can make.

8.3 Case Study of Flood at Stejaru in Romania

Romania was the most affected by the 2005 European floods, as it was faced with the most powerful and widespread floods and also the highest loss of lives, with 31 deaths. Total damages are estimated to be valued at more than 5 billion lei (€1.5 billion).

8.3.1 Description of Study Area

The Stejaru flood in Romania caused serious damage to homes and villages. The flood is happened in a large scale area. From this case study, we only choose a small area range from N46° 15' to N46° 18' and E26° 44' 24" to E26° 49' 12" as the flood simulation test. The study area of Stejaru in Romania is shown in Figure 8.22 which is attained from Google Map.

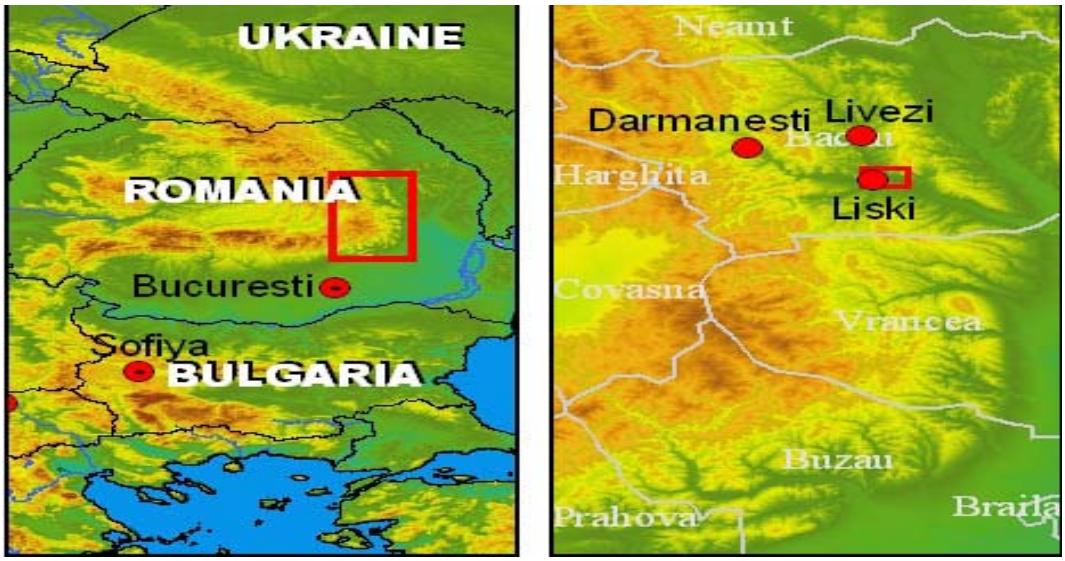


Figure 8.22 Location map of the Stejaru and the study area
(Image source from Google Map)

8.3.2 Available Data

The case of flooding happened in the city Stejaru. The grid for the analyzed area was generated by using 3D real world modelling. The texture map for the surface of Stejaru terrain is obtained from Google Earth. The analyzed domain is 5 km * 8 km [SRF]. The maximum flow is a discharge of 4000 m^3/s, while the annual medium flow is a discharge of 1500 m^3/s. The time step is 15 seconds and the total duration of simulation is 400 seconds.

8.3.3 Texture Map Analysis and Segmentation

The texture map of Stejaru is obtained from Google Earth in Figure 8.23. Before modelling and reconstructing this 3D Stejaru terrain, image analysis such as specific areas segmentation is important. By using our active contours method, the river and urban areas have been successfully recognised with different distinct colours which are shown in Figure 8.26 and Figure 29.

Figure 8.23 Texture map of Stejaru

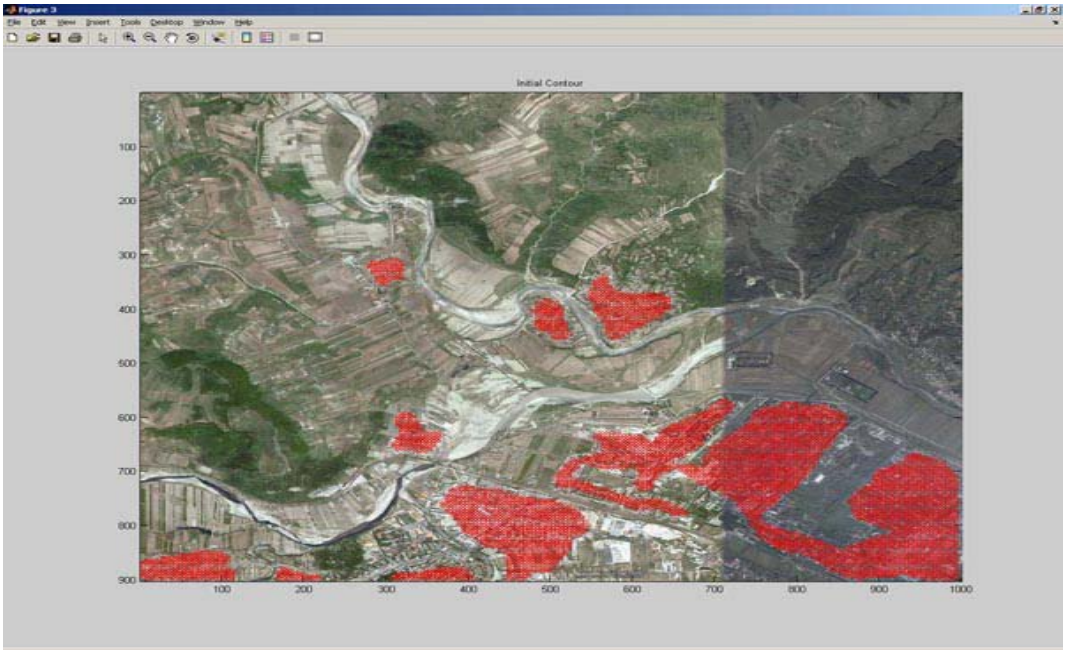


Figure 8.24 Initial active contours of urban area

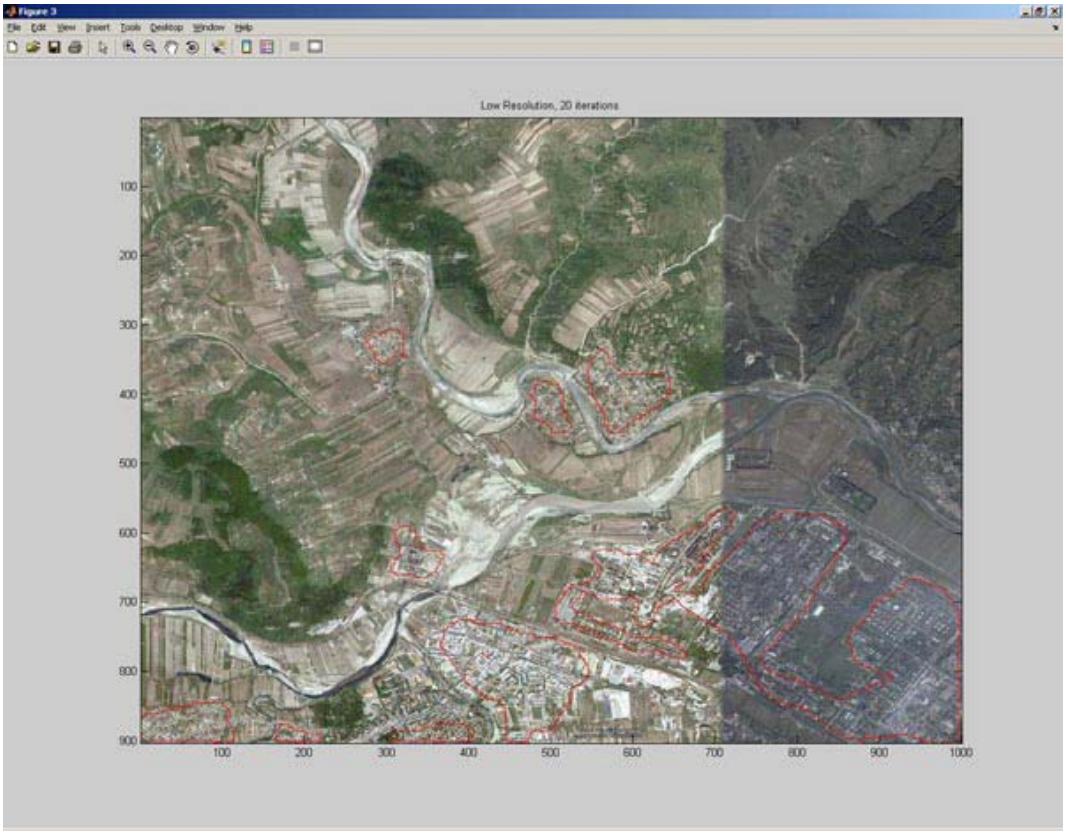


Figure 8.25 Final contours of urban area

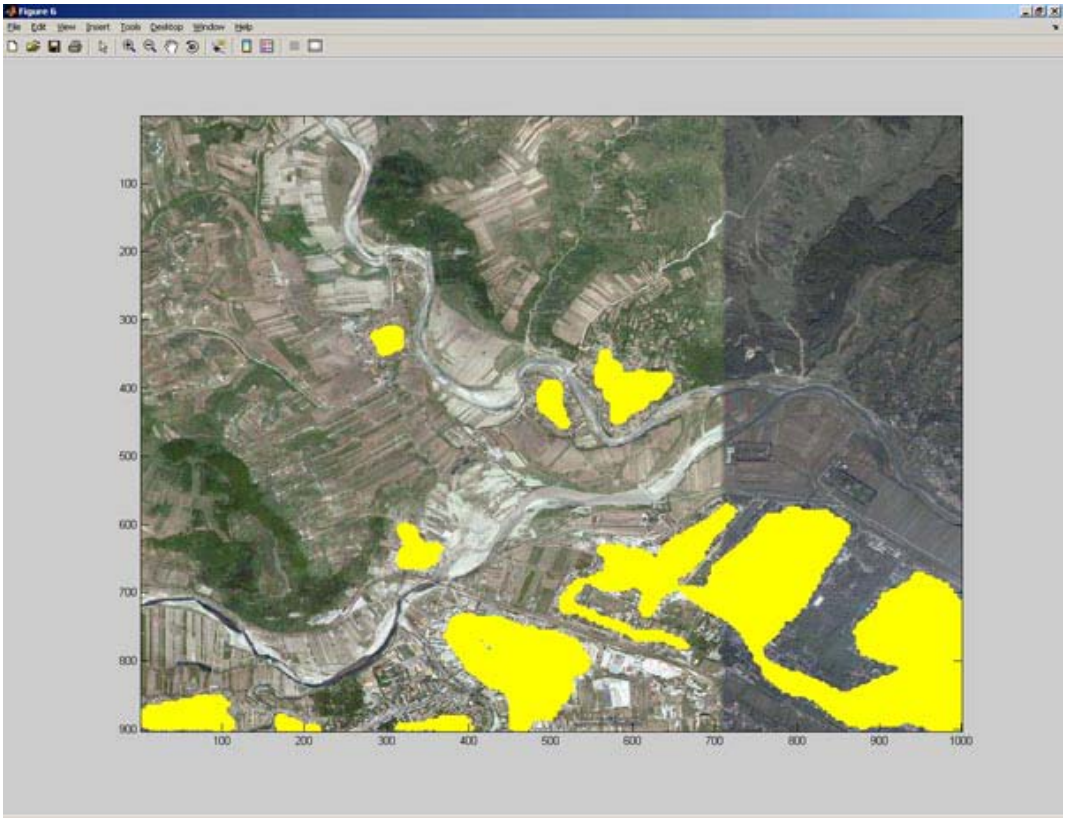


Figure 8.26 Segmented urban area with yellow colour

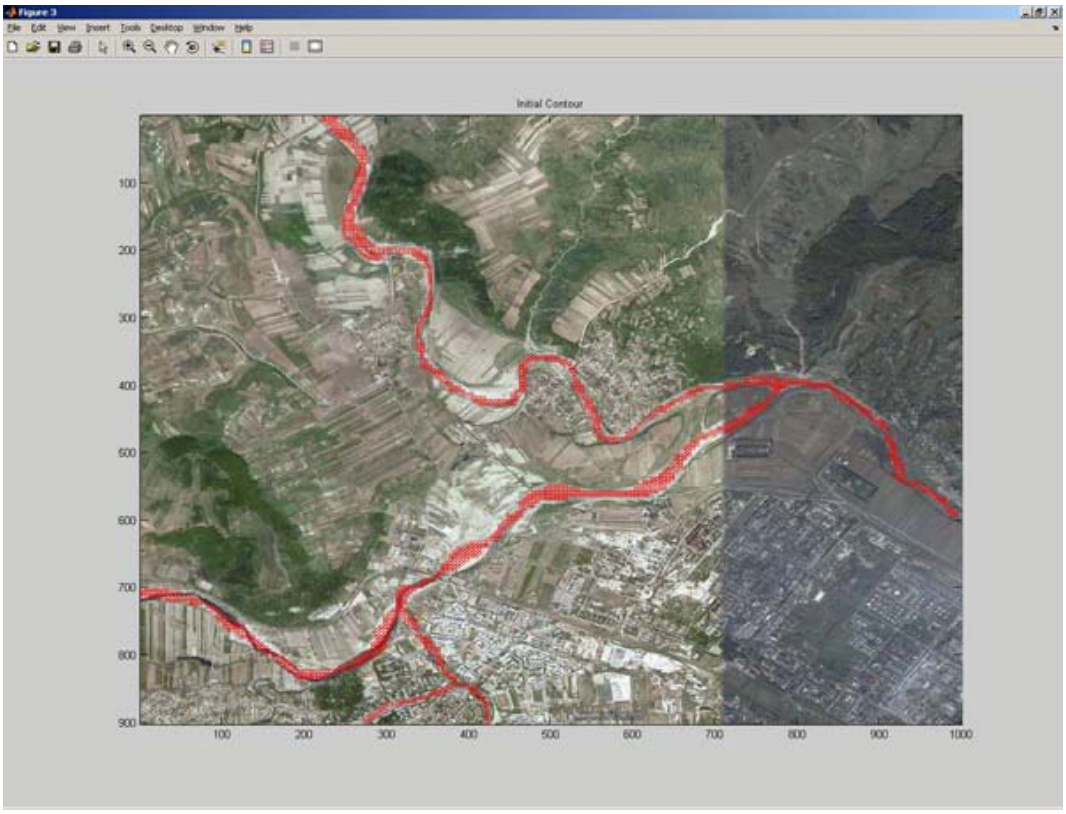


Figure 8.27 Initial active contours of river

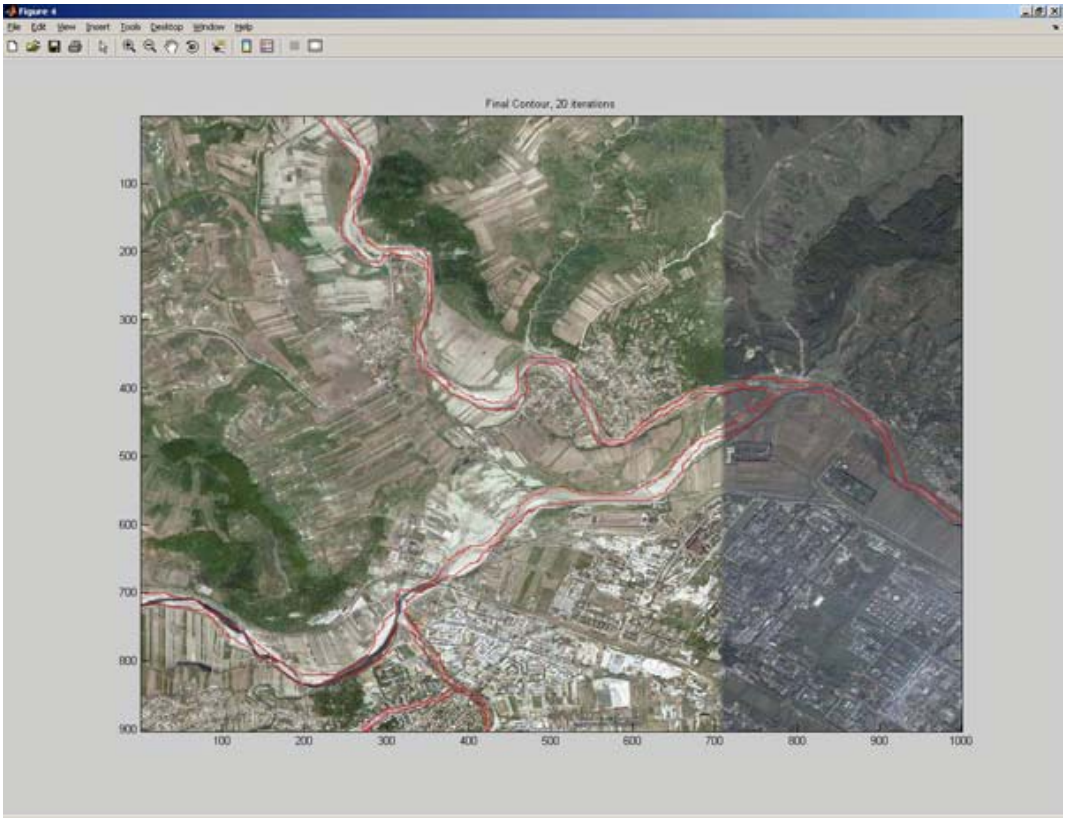


Figure 8.28 Final contours of river area

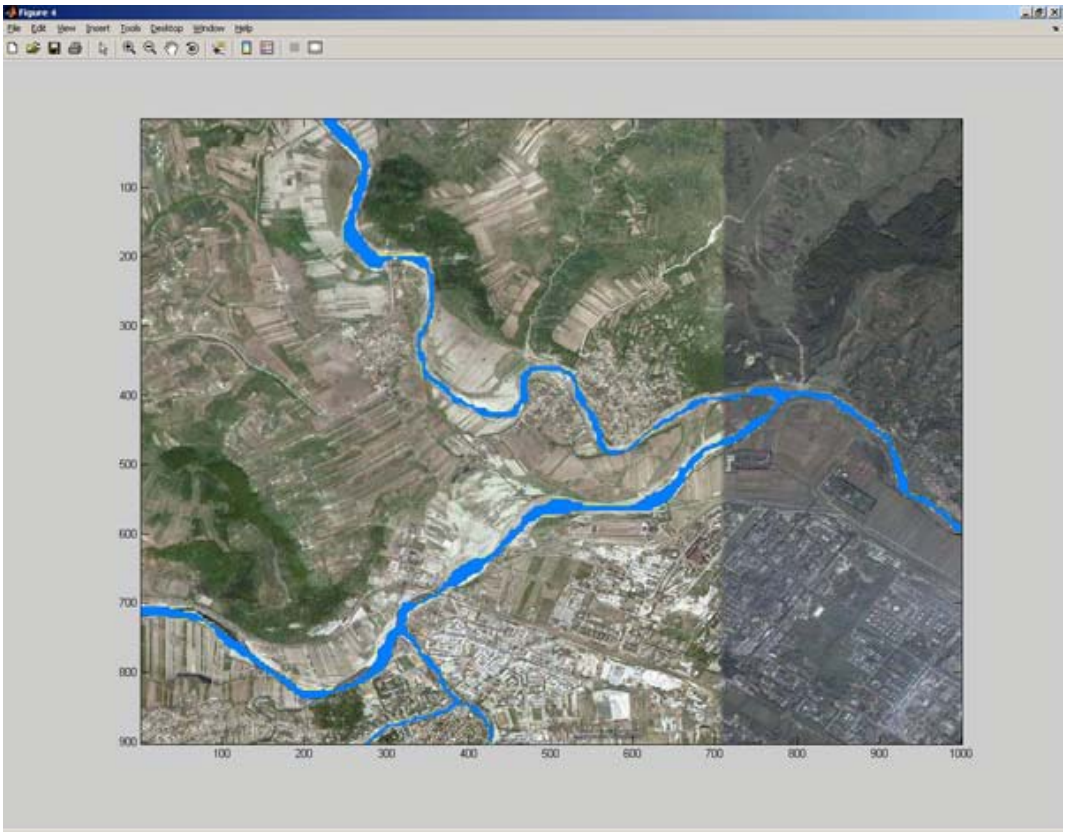


Figure 8.29 Segmented river area with cyan colour

8.3.4 Digital Terrain Modelling and Reconstruction

For modelling the 3D environment of Stejaru, we use the proposed method in chapter 3. SRTM data is used for generating 3D geometry of Stejaru and the satellite image of Stejaru is obtained from Google Earth. The generated terrain geometry is overlaid by satellite image corresponding to the same geographic area which is presented in Figure 8.30. For terrain reconstruction, at first, we need to segment the image and select the objective areas for further study.

Figure 8.31 and Figure 8.32 show the segmented river and urban areas by using our active contours method and region growing approach which is mentioned in chapter 4 and chapter 5.

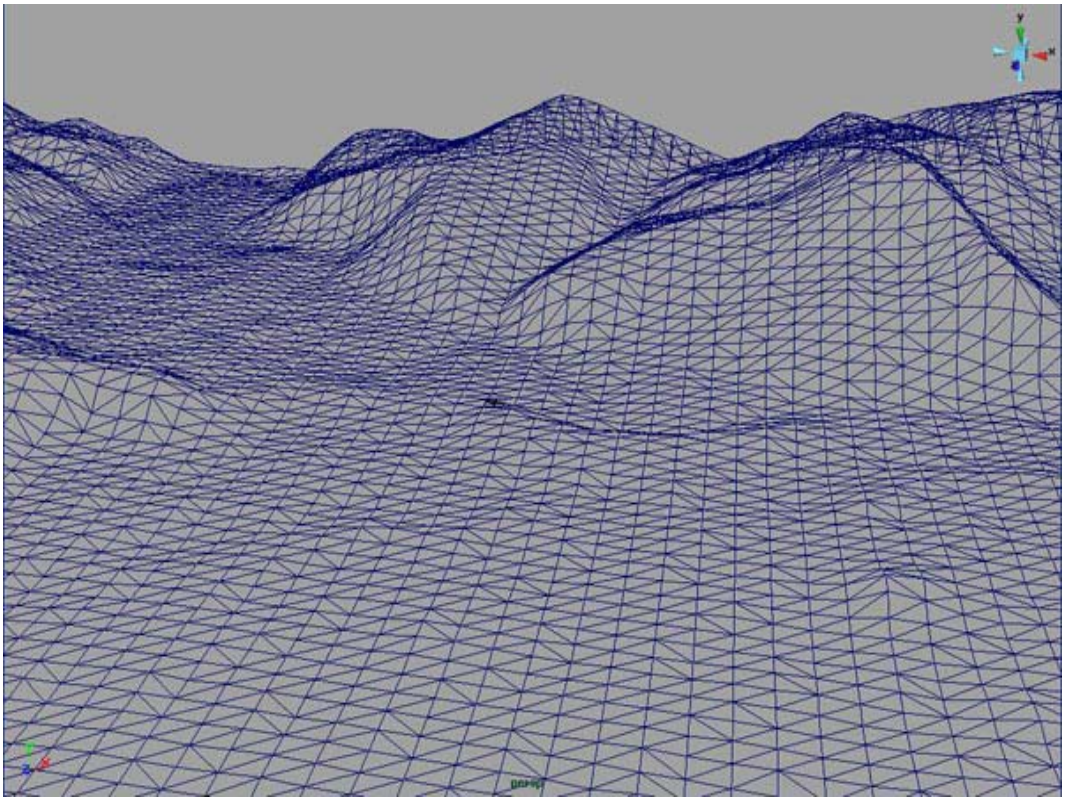


Figure 8.30 The Stejaru terrain geometry

Figure 8.31 3D environment of Stejaru terrain

Figure 8.32 3D reconstructed environment of Stejaru

Figure 8.33 The close distance view of the Stejaru region

8.3.5 Simulation Results

The test has been conducted for simulating the potential flood of the city Stejaru in Romania. This is for small scale flood simulation which contains more detailed information. The grid for the analyzed area was generated by using our 3D real world model. The analyzed domain is 5 km x 8 km. The river that flows through center of the city Stejaru is used for testing.

When the river water starts overflowing its bank, flood hazard happens. The computed water elevation surfaces at t=50s, t=200s and t=400 are presented in Figure 8.34.

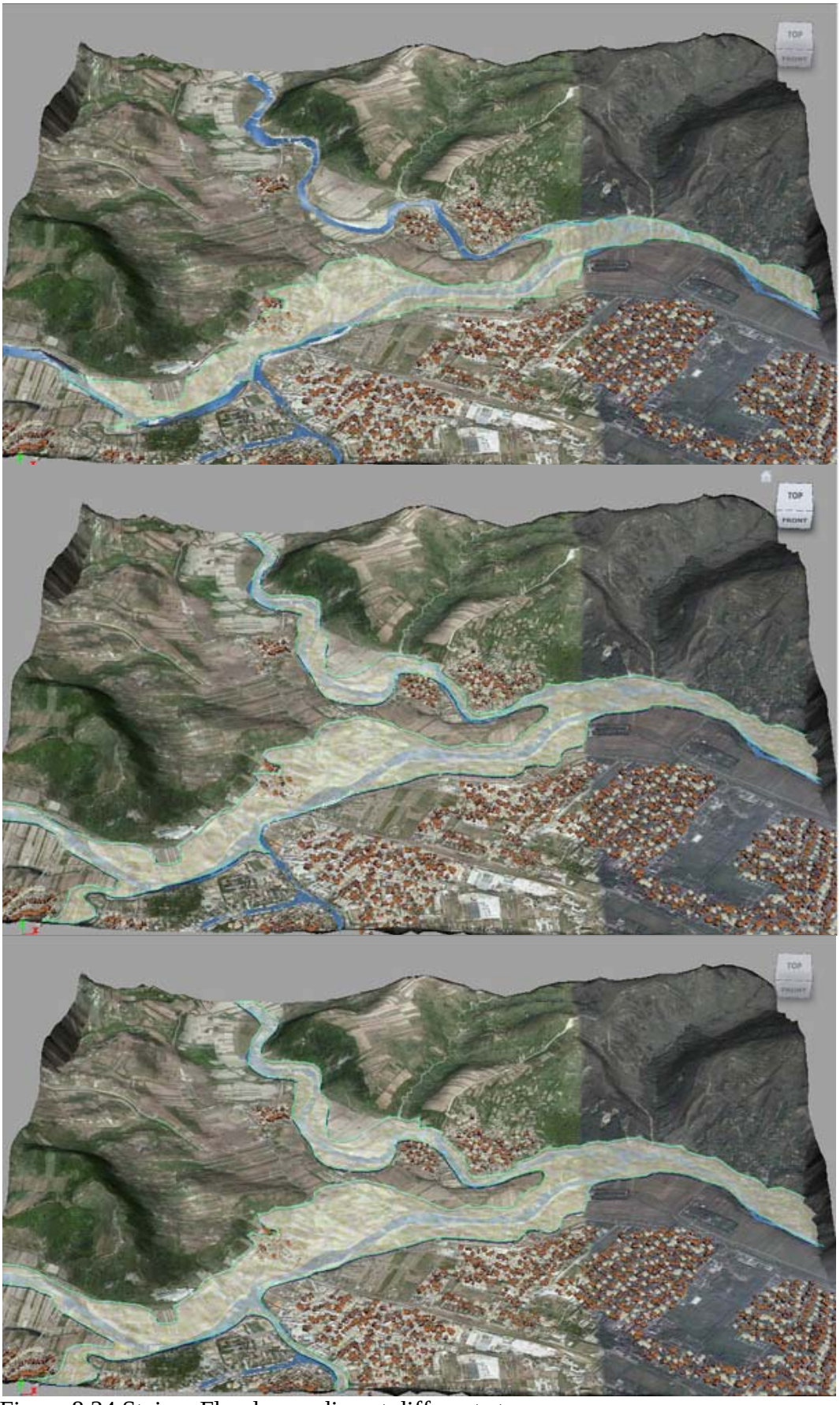


Figure 8.34 Stejaru Flood spreading at different stages

Figure 8.34 presents the eventually flooded area, while Figure 8.35 shows the real flood image of Stejaru. Compared with these two results, the simulated flood area is very close to the real inundation area.

Figure 8.35 Real flood image of Stejaru

8.3.6 Discussion

In this flood simulation test, satisfied experiment result has been achieved. At the same time, some other aspects of this work needs to be reconsidered. In the event that we can add some coefficients and infiltration influence of different types of soil in the flood spreading model construction, it can increase the quality and prediction precision of simulation process. Flood inundation area calculation depends on precision of elevation grids, the more dense grids we get, the more accurate prediction of inundation area we can make.

8.4 Case Study of London and Thames Estuary Flood

London and the Thames Estuary are the areas where the risk of flooding is particularly high. The Thames region is increasingly at risk from flooding due to higher mean sea levels, increased rainfall and tide ranges, and a greater number and intensity of storm events.

8.4.1 Description of Study Area

London and the Thames Estuary have experienced the effects of flooding in the past. Central London last flooded in 1928, when 14 people drowned. Whilst major floods in 1953 caused many areas in the outer estuary and Essex coasts to be inundated by floodwaters.

Following the disastrous events of the 1928 and 1953 floods, the UK government initiated the construction of a flood defence scheme for London throughout the 1970s and 1980s. By far the largest and most expensive part of this scheme is the Thames Barrier. The data will also be fed into a longer term project, known as Thames Estuary 2100, which aims to determine the level of flood protection needed over the next 100 years [FD]. We have been investigating the effects of a 1m-high wave travelling up the Thames, using computer simulations. The study area of London and Thames Estuary is shown in Figure 8.36 which is attained from Google Map.

Figure 8.36 Location map of London and Thames Estuary
(Image source from Google Map)

8.4.2 Available Data

The case of flooding is in the upper part of the river Thames. The grid for the analyzed area was generated by using 3D real world modelling. We prepared this topographic data from the SRTM and converted the data into the DEM, whose coordinates from N52° 22' to N52° 23' and W2° 18' to W2° 20' with 512*256 nodes [EDOS]. The texture map for the surface of London and Thames Estuary terrain is obtained from Google Earth. The analyzed domain is 3600m * 1800m. The forcing condition is a discharge of 950 m^3/s, which applied suddenly by a flood wave that moves down the centre of the city. The time step is 15 seconds and the total duration of simulation is 450 seconds.

8.4.3 Texture Map Analysis and Segmentation

The texture map of London and Thames Estuary is obtained from Google Earth in Figure 8.37. Before modelling this 3D terrain, image analysis such as

specific areas segmentation is important. By using our active contours method，one part of Thames River has been successfully recognised with cyan colour.

Figure 8.37 Texture map of one part of London and Thames Estuary

At first, we assign initial contours within the texture map based on specific characters from the river as shown in Figure 8.38. All of these initial snakes will move toward the river boundary. If one snake is getting close to the other one, then they will be automatically merged together for a big snake and so on. The final snake in texture map is shown in Figure 8.39 which is just after several iterations. Figure 8.40 presents the segmented one part of Thames River area with filled colour.

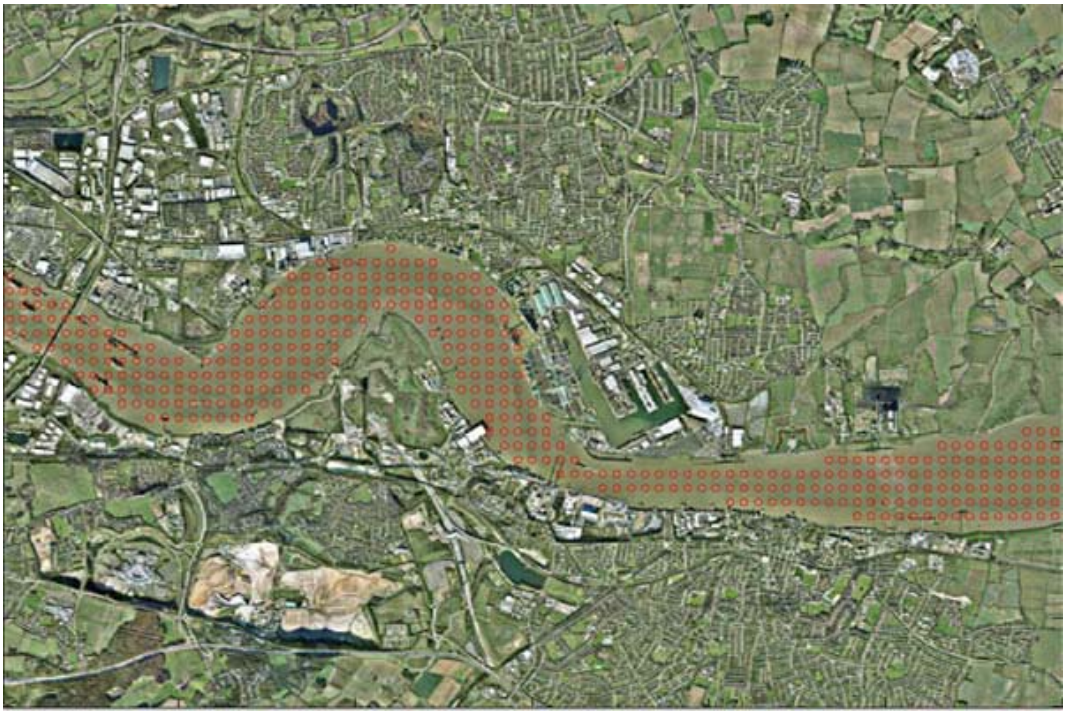


Figure 8.38 Initial active contours of one part of Thames River area

Figure 8.39 Final contours of one part of Thames River area

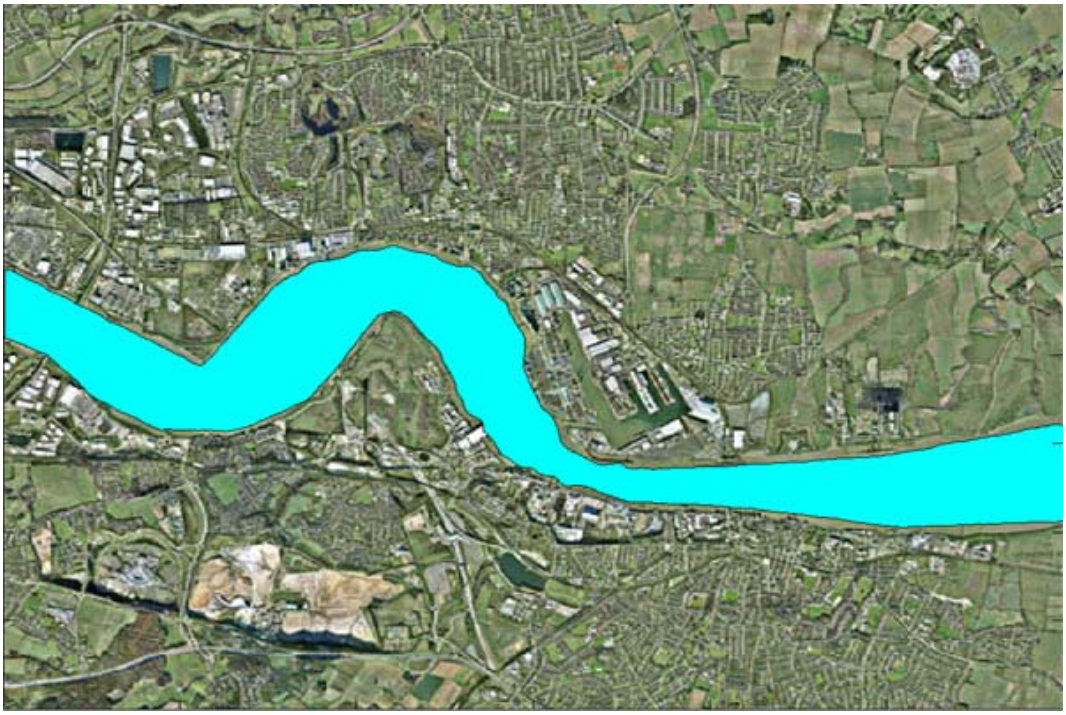


Figure 8.40 Segmented one part of Thames River area

8.4.4 Digital Terrain Modelling and Reconstruction

For modelling the 3D environment of one part of London and Thames Estuary, we use the proposed dynamic terrain generation method which is mentioned in chapter 3. SRTM data is used for generating 3D geometry of London and Thames Estuary and the satellite image is obtained from Google Earth. The generated terrain geometry is overlaid by satellite image corresponding to the same geographic area which is presented in Figure 8.41. For terrain reconstruction, at first, we need to segment the image and select the objective areas for further study. Figure 8.42 shows the one part of Thames River reconstructed environment which is a nice platform for carrying out flood simulation.

Figure 8.41 3D environment of one part of London and Thames Estuary

Figure 8.42 3D one part of Thames River reconstructed environment

8.4.5 Simulation Results

There is an interesting picture of how London would look after a 1 metre rise in sea level (Figure 8.43), which may happen over the next century according to an expert from the Potsdam Institute for Climate Impact Research. This map reveals how Westminster Abbey, the Houses of Parliament and Canary Wharf will be among the areas at risk of flooding according to a new estimate of rising sea levels.

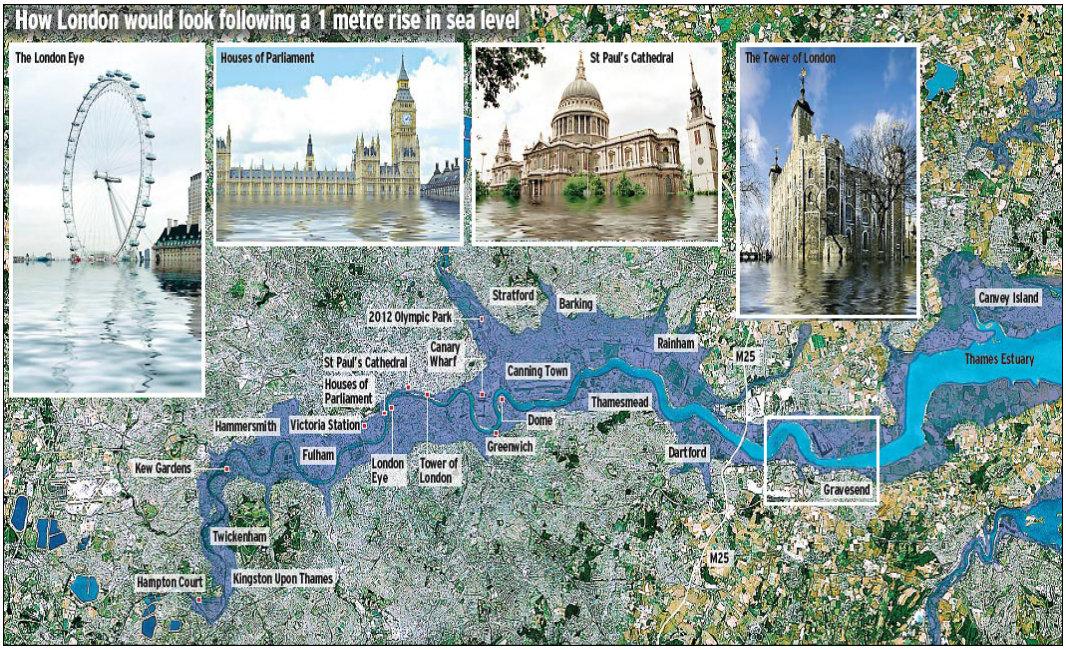
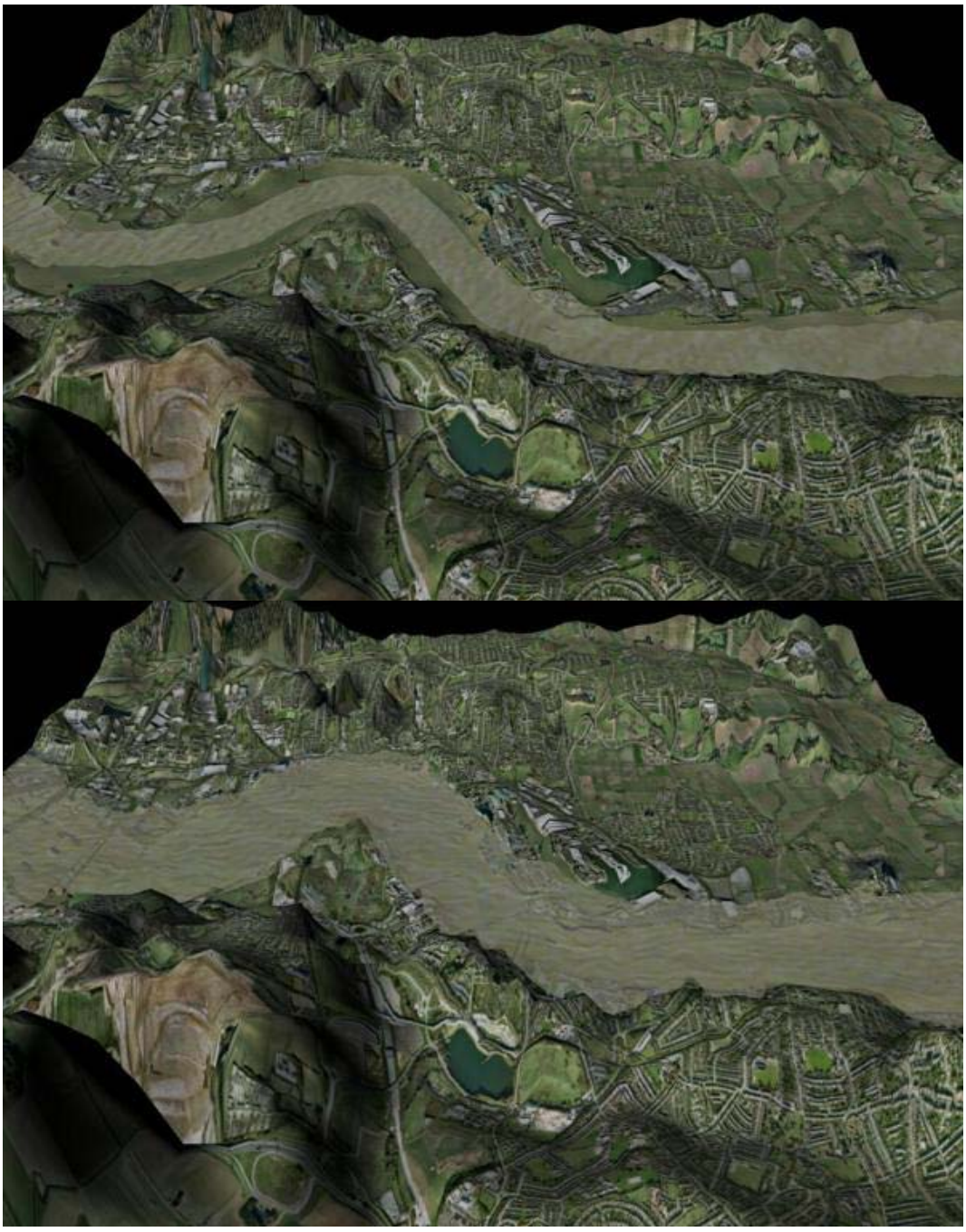


Figure 8.43 The areas where London will flood

Figure 8.44 People surprised by flash floods in West London in May 2001.

In May 2001, a flash flood hits the west London that many people were affected by this event. Figure 8.44 shows the real flood scene of that time. So it is important to carry out our research to simulate the situation when flood occurs in London and Thames River. The following figures show the inundation areas at different stages.



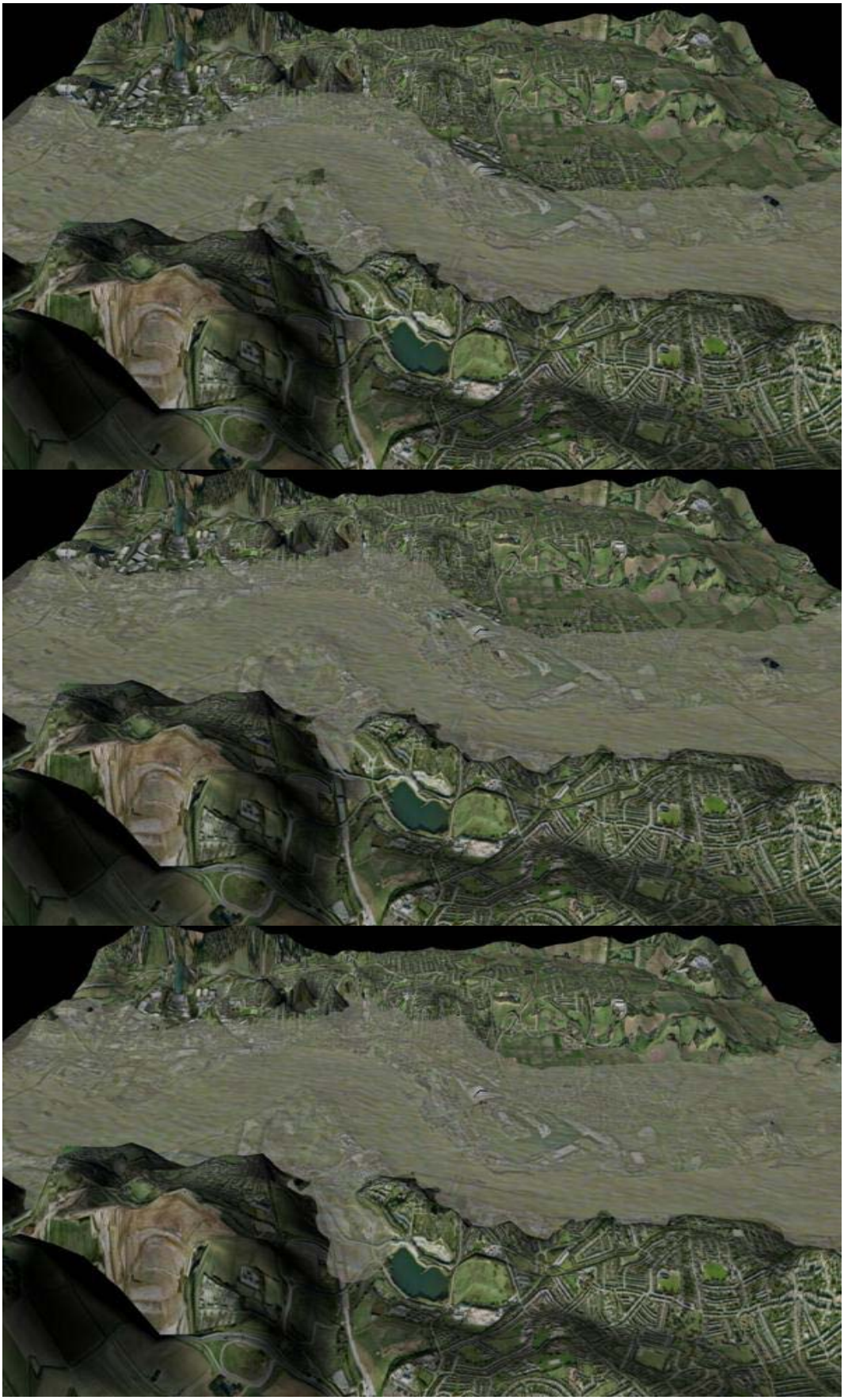


Figure 8.45 Thames flood spreading at different stages

Figure 8.45 presents the eventually flooded area, while Figure 8.46 shows the real flood image of one part of Thames taken from Figure 8.43. Compared with these two results, the simulated flood area is very close to the potential inundation area.

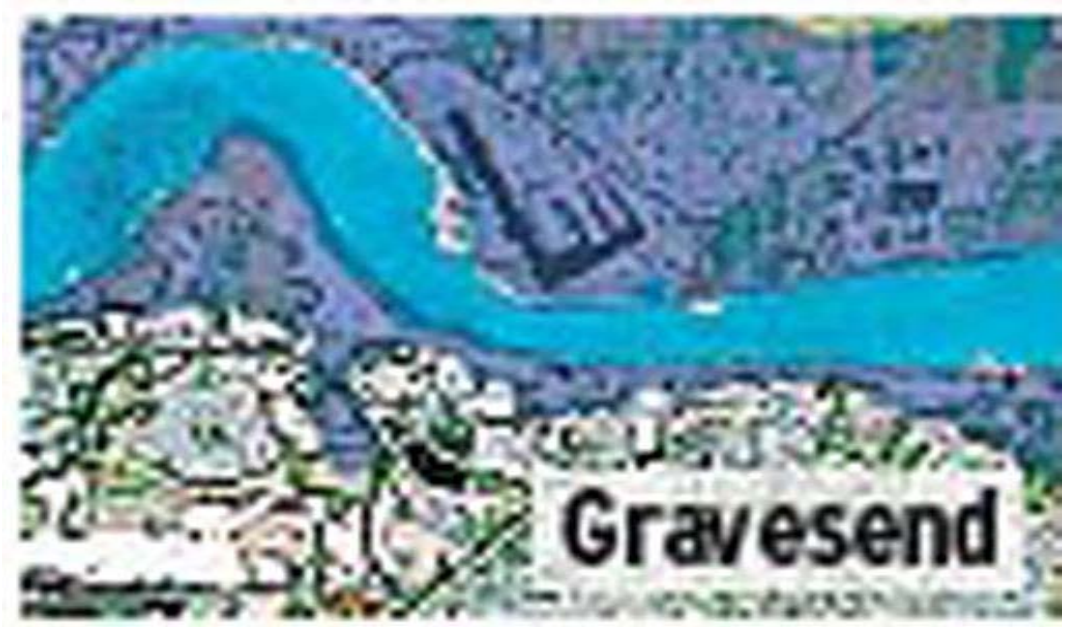


Figure 8.46 The real flood image of one part of Thames

8.4.6 Discussion

In this flood simulation test, satisfied experiment result has been achieved. The digital terrain model is constructed using real world measurement data of GIS, in terms of digital elevation data and satellite image data. The active contours-based multi-resolution vector-valued framework is used for extracting necessary objects from images. The flood spreading model is applied into the 3D digital terrain model which is useful for calculating flood inundation areas. The studies will eventually be used to determine the appropriate level of protection needed for London and the Thames Estuary for the next several years.

8.5 Case Study of Bewdley Flood

Bewdley is a lovely, largely Georgian town nestling on the banks of the River Severn in Worcestershire. The increased frequency of recent floods has highlighted the need to look urgently at the possibilities for alleviating the effects of flooding in Bewdley.

8.5.1 Description of Study Area

In the autumn of 2000, the worst flooding for over 50 years devastated the length of the River Severn, hitting Bewdley particularly badly with levels reaching 5.6 metres above summer levels. The town was extensively flooded three times in the space of six weeks. Public and political interest was raised, resulting in a renewed drive to provide a flood warning scheme.

The study area of Bewdley is shown in Figure 8.47 which is attained from Google Map.

Figure 8.47 Location map of Bewdley (Image source from Google Map)

8.5.2 Available Data

The case of flooding is in the upper part of the River Severn. The grid for the analyzed area was generated by using 3D real world modelling. We prepared this topographic data from the SRTM and converted the data into the DEM, whose coordinates from N52° 22' to N52° 23' and W2° 18' to W2° 20' with 512*256 nodes [House of Commons 2004]. The texture map for the surface of Bewdley terrain is obtained from google earth. The analyzed domain is 3600m * 1800m. The forcing condition is a discharge of 1050 m^3/s, which applied suddenly by a flood wave that moves down the centre of the city. The time step is 25 seconds and the total duration of simulation is 500 seconds.

8.5.3 Texture Map Analysis and Segmentation

The texture map of Bewdley is obtained from Google Earth in Figure 8.48. Before modelling this 3D terrain, image analysis such as specific areas segmentation is important. By using our active contours Method, the river in Bewdley has been successfully recognised.

Figure 8.48 Texture map of Bewdley

For urban area segmentation, at first, we assign initial contours within the texture map based on specific characters from urban area as shown in Figure 8.50. All of these initial snakes will move toward the river boundary. If one snake is getting close to the other one, then they will be automatically merged together for a big snake and so on. The final snake in texture map is shown in Figure 8.51 which is just after several iterations.

Figure 8.49 Segmented river in Bewdley

Figure 8.50 Initial active contours of Bewdley urban area

Figure 8.51 Final contours of Bewdley urban area

8.5.4 Digital Terrain Modelling and Reconstruction

For modelling the 3D environment of Bewdley, we use the proposed dynamic terrain generation method which is mentioned in chapter 3. SRTM data is used for generating 3D geometry of Bewdley and the satellite image is obtained from Google Earth. The generated terrain geometry is overlaid by satellite image corresponding to the same geographic area which is presented in Figure 8.52. For terrain reconstruction, at first, we need to segment the image and select the objective areas for further study. Figure 8.53 shows the Bewdley urban area reconstructed environment which is a nice platform for carrying out flood simulation. We also provide a close view of Bewdley reconstructed environment (Figure 8.54).

Figure 8.52 3D environment of Bewdley

Figure 8.53 3D Bewdley reconstructed environment

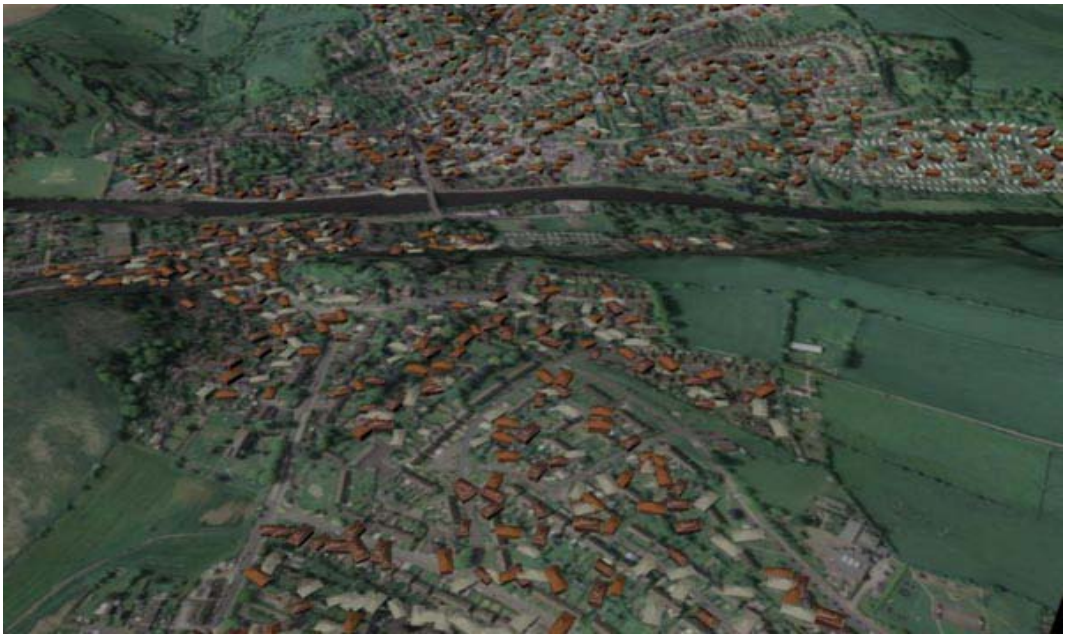


Figure 8.54 The close distance view of Bewdley reconstructed environment

8.5.5 Simulation Results

There is a real picture (Figure 8.55) of 2007 Bewdley flood when the River Severn overflowed its banks. In total around 200 properties were flooded in the Bewdley area. The flood inundation map (Figure 8.56) of Bewdley is also listed for comparison with our simulation results.

Figure 8.55 Bewdley suffered from flood in June 2007.

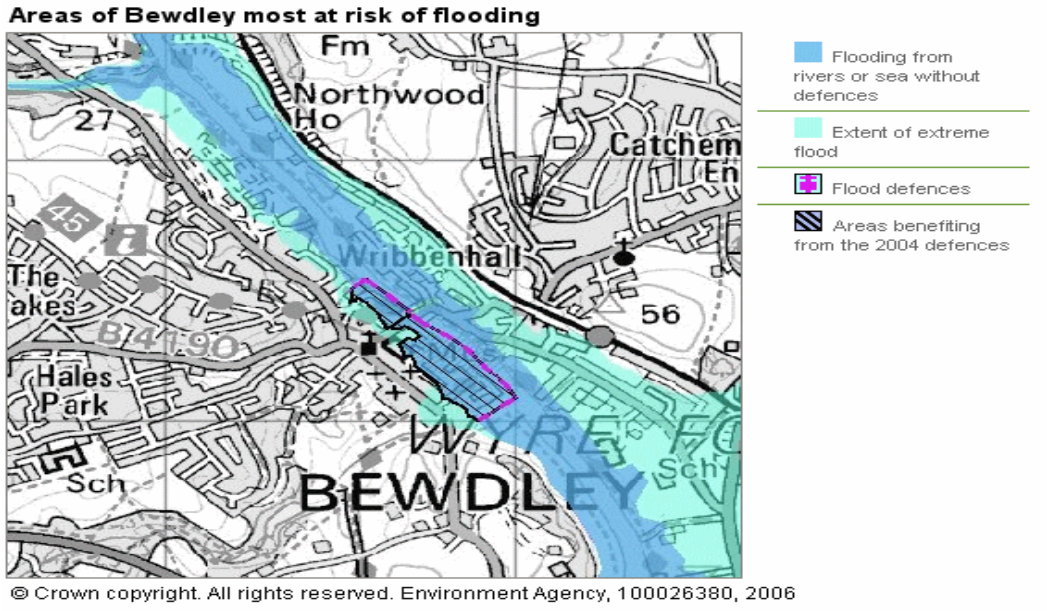


Figure 8.56 Bewdley flood inundation map.

When the river water starts overflowing its bank, flood hazard happens. The computed water elevation surfaces at t=50s, t=150s, t=250s, t=350s and t=500s are presented in Figure 8.57.

Figure 8.57 Bewdley flood spreading at different stages

8.5.6 Discussion

In this flood simulation test, satisfied experiment results have been achieved. The digital terrain model is constructed using real world measurement data of GIS, in terms of digital elevation data and satellite image data. This test is based on 3D environmental reconstruction system which consists of environmental image segmentation for object identification, a shape match method and an intelligent reconstruction system. The active contours-based multi-resolution vector-valued framework and the multi-seed region growing method are both used for extracting necessary objects from images. The studies will eventually be used to determine the appropriate level of protection needed for Bewdley for the next several years.

8.6 Case Study of Tewkesbury Flood

Tewkesbury is a town of Gloucestershire in England. It stands at the confluence of the River Severn and the River Avon, and also minor tributaries the Swilgate and Carrant Brook. The area around Tewkesbury is frequently affected by flooding. In general such flooding causes little damage to property as the town is surrounded by large areas of floodplain which restrict urban development and the ability for the town to spread. However, extreme flooding events have caused damage to property and affected transport links, the most significant events occurring in 1947, 1960 and 2007.

8.6.1 Description of Study Area

In July 2007, River Severn and the River Avon which meet at Tewkesbury were overwhelmed by the volume of rain that fell in the surrounding areas on one day, Friday 20th July. Tewkesbury was completely cut off with no road access, parts of the town were under around 3 feet (0.9 m) of water and flood waters entered Tewkesbury Abbey for the first time in 247 years [FASW]. Tewkesbury's Mythe Water Treatment Works were flooded, threatening drinking water supplies to 350,000 people, and Severn Trent Water warned that treated water would run out by early Sunday evening in Tewkesbury. Combined military and civil emergency services tried to stop floods reaching the Walhamelectricity substation in Gloucester supplying half a million people.

8.6.2 Available Data

The case of flooding is in the upper part of the River Severn. The grid for the analyzed area was generated by using 3D real world modelling. We prepared this topographic data from the SRTM and converted the data into the DEM, whose coordinates from N52° 21' 45" to N52° 21' 58" and W2° 18' 35" to W2° 18' with

512*256 nodes .The texture map for the surface of Tewkesbury terrain is obtained from Google Earth. The analyzed domain is 650m * 325m. The forcing condition is a discharge of 820 m^3/s, which applied suddenly by a flood wave that moves down the centre of the city. The time step is 25 seconds and the total duration of simulation is 650 seconds.

8.6.3 Texture Map Analysis and Segmentation

The texture map of local region of Tewkesbury is obtained from Google Earth in Figure 8.58. Before modelling this 3D terrain, image analysis such as specific areas segmentation is important. By using our active contours method, the whole environmental image has been segmented and identified with several distinct colours. Figure 8.59 shows the processed image with different recognised specific areas: pink indicates the street area, yellow displays the houses, blue shows the river area, cyan represents garden areas, green is for forest area and light green is for grassland area.

Figure 8.58 Texture map of local region of Tewkesbury

Figure 8.59 Segmented image of local region of Tewkesbury

8.6.4 Digital Terrain Modelling and Reconstruction

For modelling the 3D environment of local region of Tewkesbury, we use the proposed dynamic terrain generation method which is mentioned in chapter 3. SRTM data is used for generating 3D geometry of Tewkesbury and the satellite image is obtained from Google Earth. The generated terrain geometry is overlaid by satellite image corresponding to the same geographic area which is presented in Figure 8.60. For terrain reconstruction, at first, we need to segment the image and select the objective areas for further study. Figure 8.61 shows the local region of Tewkesbury reconstructed environment which is a nice platform for carrying out flood simulation.

Figure 8.60 3D environment of local region of Tewkesbury

Figure 8.61 3D reconstructed environment of local region of Tewkesbury

8.6.5 Simulation Results

There is a real picture (Figure 8.62) of 2007 Tewkesbury flood when the River Severn and the River Avon overflowed its banks. In total around 2000 properties were flooded in the Tewkesbury area.

Figure 8.62 Tewkesbury suffered from flood in July 2007

In Figure 8.63, we have computed flood flow direction and inundatiuon areas at different stages by using the flood spreading model. This flood model based on standard gradient descent method for energy minimization can predict the flood inundation areas precisely.

Figure 8.63 Computed Flood Areas by Flood Spreading model

When the river water starts overflowing its bank, flood hazard happens. The computed water elevation surfaces at t=75s, t=150s, t=250s, t=450s and t=650s are presented in the following Figures.

Figure 8.64 Local region of Tewkesbury flood spreading at different stages

We also provide a close distance view of Local region of Tewkesbury flood at t=650s (Figure 8.65).

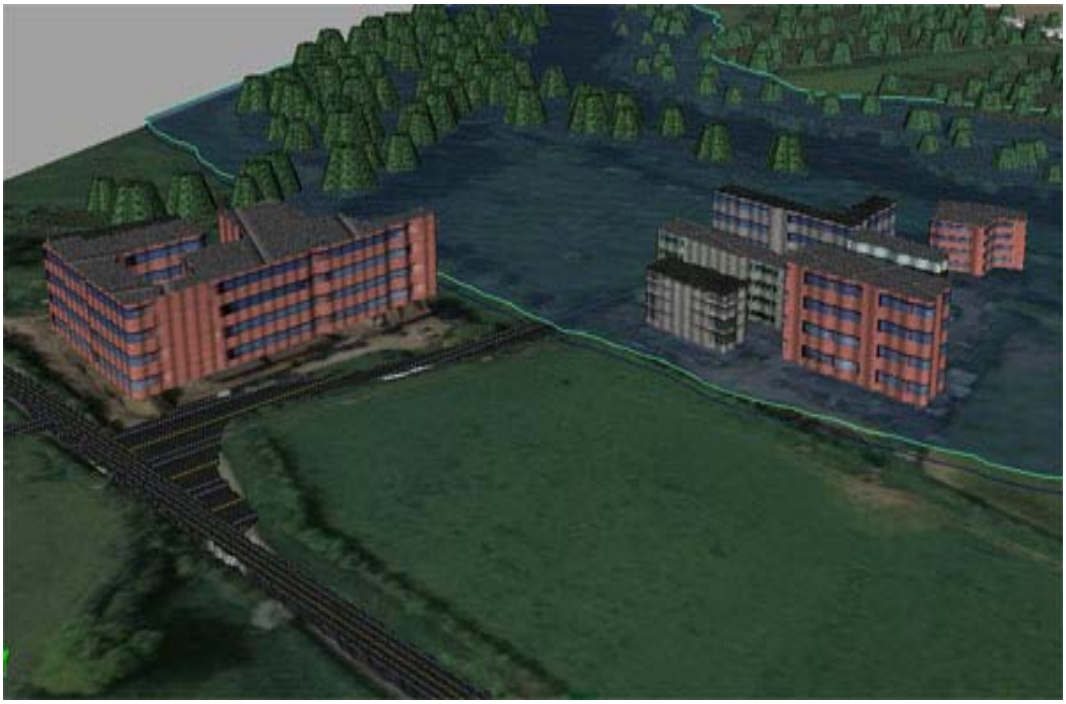


Figure 8.65 The close distance view of Tewkesbury flood spreading at t=650s

8.6.6 Discussion

In this flood simulation test, satisfied experiment result has been achieved. This test is based on the 3D environmental reconstruction system which consists of environmental image segmentation for object identification, a shape match method and an intelligent reconstruction system discussed in previous chapters. The active contours-based multi-resolution vector-valued framework and the multi-seed region growing method are both used for extracting necessary objects from images. The flood spreading model is useful for calculating the flood inundation area. The studies will eventually be used to determine the appropriate level of protection needed for Tewkesbury for future flood warning.

8.7 Conclusion

The performance of the flood spreading model and the finite volume hydrodynamic model in flood simulation are tested in this chapter. The outcomes of the model show that the model can simulate the flood extent along the main river channels and the floodplain with a reasonable level of accuracy. The simulated water levels along the River Thames, River Severn and Stejaru show consistency with the real flood situation which gives better results than a 2D modelling approach. This 3D solution scheme of the flood spreading model is suitable for fast computation, which is essential for an early-warning system. The finite volume hydrodynamic model is useful for forecasting flood in local and dense areas more precisely. Further research is in progress to improve the model so as to eliminate the accumulated errors through a flood spread in each of the time steps.

Chapter 9 Conclusions and Future Work

There is a growing need for geographical information and processing applications especially for large scale geographical areas in last few years. Most of the existing research on environmental reconstruction has focussed on creating aesthetically pleasing models for virtual tours and other entertainment purposes. The algorithms and solutions proposed in this work will be applicable to a variety of applications including emergency simulation, flood hazard warning, urban planning and change monitoring and environmental resource management.

9.1 Conclusions

This thesis has concentrated on developing an efficient flood hazard warning system by using 3D large scale reconstructed environment and dynamic flooding models. Evaluations of the system are carried out through analysis, simulations and prototyping. The following conclusions can be drawn:

1) A new dynamic terrain modelling method is proposed with four major components for generating 3D real world. i) A nona tree space partitions (NTSP) algorithm for real-time terrain generation capable of dealing with very large GIS data has been presented. The centre of a local region has been assigned as a root node, which stores all the vertices of the local region. This arrangement makes it much easier to access GIS data for constructing the world in real time, and for navigation towards neighbouring regions. The root node is divided into nine parts represented by the corresponding nodes, this subdivision will go on until a threshold is reached. ii) The pyramidal data arrangement structure is used to deal with detailed real world when an observer moves close to the earth surface. At

layer 1, a large scale terrain is reconstructed using a low resolution data set. A high resolution model is created in layer 2 with 9 region elements, Layer 3 representing 81 region elements with higher resolution, and so on. iii) A high resolution urban environment needs more grids in comparison to a low resolution environment if both environments are in the same level of details. The linear interpolation of the terrain data is proposed for creating a smoother surface with the method using curve interpolations. iv) At present time, almost all GIS systems are based on a Gaussian coordinate system, which is a Gauss plane, a linear space. Since the Earth space is nonlinear, the new map projection is proposed which can portray the surface of the earth or a portion of the earth on a flat surface. The new dynamic terrain modelling method offers the following features:

- It can perform an out-of-core rendering and to deal with massive GIS terrain data which is easy to be implemented using hardware memory.
- It creates a smoother surface by curve interpolation of terrain data which solves the problem of low resolution terrain environment.
- It can process and present the detailed terrain information with high resolution data set through in-core pyramidal data arrangement structure.
- It offers a new map projection which can portray the surface of the earth on a flat surface instead of linear space.

2) In order to reconstruct 3D large scale environment, image segmentation is very important. A novel multi resolution vector valued framework for image segmentation based on active contours has been proposed. This framework consists of vector valued scanning algorithm, the geometric active contours model, low resolution segmentation with elevation data and high resolution optimization. Vector valued scanning algorithm is for assigning initial snakes

depends on minimum, maximum, average and standard deviation values together with texture features from each sampling cell. Active contours are used to move toward the object boundaries with fast speed. Low resolution segmentation with elevation data is adopted to get the coarse areas of target object. After achieving boundary at low resolution, the next step is passing extracted contour to high resolution for optimization. For environmental objects identification, a novel shape match method is proposed from satellite images in high resolution. At first, some basic types of primitive for environmental objects are defined. By using our geometric active contours model, the boundaries of environmental objects can be extracted and saved in binary maps. Then the intelligent shape match method is used to analysis the binary maps and choose the most similar type of primitive to model the environmental object. It offers the following advantages from AC based image segmentation and intelligent shape match based environmental objects identification:

- A vector-valued scanning algorithm is discussed which can automatically assign initial snakes as if their specific values inside the calculated vector of image features.
- An active contours model is used to detect object boundaries. This model can be easily incorporated into various prior knowledge such as shape and region distribution. The final contours are quite regular, which are convenient for further applications, such as shape analysis, classification, and recognition.
- The combined low resolution segmentation and high resolution optimization algorithm allows for a rapid evolution of the contour and a convergence to its final configuration with a small number of iterations.

- Environmental objects can be identified rapidly by intelligent shape match method which is useful for finding primitives which contain similar parts and counting the number of similar portions automatically.

3) A new strategy is proposed to design the 3D environmental reconstruction system. For large scale reconstruction, the geometric active contours model and multi seed region growing method are used to segment the images of large scale environment; the intelligent reconstruction system is used to automatically recreate different areas such as sea region, river region, forest region and residential region. For small scale detailed reconstruction, the intelligent shape match method is employed to rebuild environmental objects with high speed. Environmental objects will be detected through our geometric active contours model. The main advantages of our 3D environmental reconstruction system are:

- It automatically identifies target areas based on our geometric active contours model and multi seed region growing algorithm.
- Environmental objects can be recreated with high speed by our shape match method in small scale detailed environment.
- It can reconstruct 3D large scale environment intelligently based on specific segmented areas.

4) As one application of our 3D environmental reconstruction system, a new dynamic simulation scheme for flood hazard warning has been proposed. Flooding models are core parts of this scheme. Two new general flood simulation models have been developed: a) Flood spreading model; b) The finite volume hydrodynamic model based on shallow water equations. Flood spreading model consists of four components: (1) Flooding image spatial segmentation (2) Water level calculation process. (3) Standard gradient descent method for energy

minimization. (4) Flood region search and merge process. The first model is built based on the flood contours energy minimization. Standard gradient descent calculation in terrain elevation map is applied into the model as well which provides a fast and efficient flood prediction. The second model is built from 2D shallow water equations which can provide more detailed flood information and behaviour, and also can be used with the first model. Both flood models utilise the digital terrain model which is built automatically from ROAM and NineTreeNodes algorithms mentioned in the previous chapters. The simulation result provides supporting decisions information for flood disaster defence and basic data for flood hazard evaluation. The proposed scheme has the following properties:

- It is capable of realistically visualising the flooding in geometrically real 3D environments. Besides, it can handle dynamic flood behaviour in real time.
- Flood spreading model is useful for large scale flood simulation which includes flooding image spatial segmentation, water level calculation process, standard gradient descent method for energy minimization and flood region search and merge process.
- The finite volume hydrodynamic model based on shallow water equations is useful for urban area flood simulation. Finite volume methods combine the simplicity of finite difference methods with the geometric flexibility of finite element methods. The set of shallow water equations is hyperbolic and therefore it has an inherent directional property of wave propagation for estimating the normal flux through each cell interface which is a key problem in FVM.

- Our method has faster speed and greater accuracy of simulating inundation area in comparison to the conventional flood simulation models.

5) For validating the performances of our flood spreading model and the finite volume hydrodynamic model, several flood simulation tests have been conducted. The studies achieved two main objectives: implementing a flood spreading process within 3D environment for predicting potential inundation area precisely; providing useful information to make a right decision when a real flood occurs.

The five case studies are listed below which offers the following features:

- Case 1 is for simulating tidal flooding at Chepstow in Wales, which tests the urban flood situation. The 3D environment of Chepstow is modelled and reconstructed separately. Flood spreading process with inundation extent map at different time is shown with an easy to use interface.
- Case 2 is selected to test the capability of this simulation model to deal with real life problems with complicated topography. The study area has been chosen at Stejaru in Romania. The simulated flood area is very close to the real inundation area which validates that our model is robust and computationally efficient.
- Case 3 is for simulating a potential flood of the London and Thames Estuary at UK. The simulated water levels along the River Thames show consistency with the real flood situation which gives better results than a 2D modelling approach.

- Case 4 is for simulating the flood affecting the city of Bewdley at UK. The flood spreading process is shown within different time and the flood inundation area is presented for further disaster assessment.
- Case 5 is for simulating a potential flood of a region of Tewkesbury in England. The flood spreading process is shown during different time and the small scale detailed inundation area is presented for further disaster evaluation.

9.2 Future Work

We propose a novel dynamic simulation scheme for flood hazard warning. The work consists of three main parts: digital terrain modelling; 3D environmental reconstruction and system development, flood simulation models. This work also unveils a number of new topics worth of further research efforts. The future research directions are addressed below for:

- Combining the environmental objects identification procedure with the correlation-based terrain extraction system for automatic reconstruction. The two techniques clearly complement each other: the terrain extraction system will be used to determine a digital elevation map upon which the volumetric environmental models will sit, and the environmental objects identification procedure will be used to identify man-made objects occlusion boundaries. A tighter coupling of the two systems, where an initial digital elevation map is used to focus attention on distinctive humps that may be buildings, or where correlation-based terrain extraction techniques are applied to building rooftop regions to identify

fine surface structure like roof vents and air conditioner units, may also be investigated.

- Incorporating the flood spreading model and hydrodynamic model into the decision-making system. A decision-making approach to evacuation is necessary and important that aims to minimize loss of life and disruptions to communities through identification of the evacuation strategy that has the maximum expected value based on currently forecasted conditions. We want to propose a general idea about evacuation: The expected value of a given action per unit of population will be calculated by integrating over the range of forecasted discharge rates, and the danger to human life associated with each rate. This expected value will vary depending on the geographical characteristics of each location being considered and the choice of action (evacuates or don't evacuate) taken. The decision model considers a trade-off between the costs to a community of issuing false alarm evacuations, and the risk of loss of life in the event where an evacuation order is not given. Once the expected value of action evacuates becomes greater than a calculated value, evacuation becomes optimal. As the potential inundation area can be simulated in our flood spreading model and the finite volume hydrodynamic model, the inundation risk level is assigned in real time to each area within the target watershed, the optimal evacuation path can be rapidly identified from a number of alternatives by selecting that route which avoids areas at high risk.
- Applying realistic virtual 3D reconstructed environment into mobile navigation. This solution can be divided into two parts: 1) It will use the

combination of aerial images, terrestrial photographic input and 2D ground maps in conjunction with a user-friendly customized interface allowing for automatically and interactively generating accurate geo referenced textured 3D environment from our proposed model; 2) In the next section of the work the geo-referenced 3D geographical areas produced will be imported in a mobile virtual reality visualisation engine, called Virtual Navigator, specifically created for the purposes of urban navigation, way finding and exploration. This engine, operating on a PDA, will be capable of reading 3D visualization file formats and includes a front-end user centred interface for fully immersive virtual 3D navigation by manipulating 6 degrees of freedom such as observer location in three dimensions (x, y, z) plus orientation (yaw, roll and pitch).

Appendix

A.3D Real World Modelling Code

```
/*****************************************************************************/

// Application:    3D Real World Modelling
// Author :        Chen Wang
// Main Program Include General System Parameters and Libraries
// Developed by using C++ and OpenGL

/*****************************************************************************/

#include <windows.h>
#include <gl\gl.h>
#include <gl\glu.h>
#include <stdio.h>
#include "terrain.h"
#include "engine.h"

HDC             hDC=NULL;           // Private GDI Device Context
HGLRC           hRC=NULL;                   // Permanent Rendering Context
HWND            hWnd=NULL;                  // Holds Our Window Handle
HINSTANCE       hInstance;              // Holds The Instance Of The Application

// Window's size
int g_WinWidth = 800,
        g_WinHeight = 600;

CFlexEngine *g_Engine;
int Tsize_flag=1;

// Functions' prototypes
LRESULT   CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM); // Declaration For WndProc
GLvoid KillGLWindow(GLvoid);
void Quit(int Message);

// Flags
bool g_ToggleFullscreen;
bool g_Fullscreen;
bool modify=true;
/*****************************************************************/
void InitScene()
{
        g_Fullscreen = false;
        g_ToggleFullscreen = false;

        g_Engine = new CFlexEngine(hWnd);
        g_Engine->Update();

}
/*****************************************************************/
int DrawGLScene(GLvoid)                                         // Here's Where We Do All The Drawing
{
        glClear(GL_COLOUR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);  // Clear Screen And Depth Buffer
        glLoadIdentity();                                       // Reset The Current Modelview Matrix

        g_Engine->UpdateNode();                                  // for dynamic terrain generation

        g_Engine->Update();

        g_Engine->RenderScene();

        return TRUE;                                                    // Everything Went OK
}

/*****************************************************************/
void ResizeGLScene(int width, int height)
```

```
{
     if (height==0) height=1;

     g_WinHeight = height;
     g_WinWidth = width;

     glViewport(0, 0, width, height);     // Reset The Current Viewport
}

/*********************************************************************/

// Frees the used structures, kills the window and quit

void Quit(int Message)
{
     delete g_Engine;


     KillGLWindow();
     exit(Message);

}

/*********************************************************************/

GLvoid KillGLWindow(GLvoid)
{
     if (g_Fullscreen)
     {
          ChangeDisplaySettings(NULL,0);         // If So Switch Back To The Desktop
          ShowCursor(TRUE);                       // Show Mouse Pointer
     }

     if (hRC) // Do We Have A Rendering Context?
     {
          wglMakeCurrent(NULL,NULL);
          wglDeleteContext(hRC);
          hRC=NULL;
     }

     ReleaseDC(hWnd,hDC);

     if (hWnd) DestroyWindow(hWnd);
     UnregisterClass("OpenGL",hInstance);
}

/*********************************************************************/

/*   This Code Creates Our OpenGL Window.  Parameters Are:                         *
 *   title           - Title To Appear At The Top Of The Window                    *
 *   width           - Width Of The GL Window Or Fullscreen Mode                   *
 *   height               - Height Of The GL Window Or Fullscreen Mode             *
 *   bits            - Number Of Bits To Use For Colour (8/16/24/32)          *
 *   fullscreenflag  - Use Fullscreen Mode (TRUE) Or Windowed Mode (FALSE)     */



BOOL CreateGLWindow(char* title, int bits)
{
     GLuint          PixelFormat;         // Holds The Results After Searching For A Match
     WNDCLASS   wc;                            // Windows Class Structure
     DWORD      dwExStyle;                // Window Extended Style
     DWORD      dwStyle;                  // Window Style
     RECT       WindowRect;          // Grabs Rectangle Upper Left / Lower Right Values
     WindowRect.left=(long)0;             // Set Left Value To 0
     WindowRect.right=(long)g_WinWidth;   // Set Right Value To Requested Width
```

```
	WindowRect.top=(long)0;                    // Set Top Value To 0
	WindowRect.bottom=(long)g_WinHeight;// Set Bottom Value To Requested Height

	hInstance  = GetModuleHandle(NULL);              // Grab An Instance For Our Window
	wc.style        = CS_HREDRAW | CS_VREDRAW | CS_OWNDC; // Redraw On Size, And Own DC For Window.
	wc.lpfnWndProc  = (WNDPROC) WndProc;         // WndProc Handles Messages
	wc.cbClsExtra   = 0;                            // No Extra Window Data
	wc.cbWndExtra   = 0;                            // No Extra Window Data
	wc.hInstance    = hInstance;                         // Set The Instance
	wc.hIcon        = LoadIcon(NULL, IDI_WINLOGO);             // Load The Default Icon
	wc.hCursor = LoadCursor(NULL, IDC_ARROW);            // Load The Arrow Pointer
	wc.hbrBackground     = NULL;                         // No Background Required For GL
	wc.lpszMenuName = NULL;                         // We Don't Want A Menu
	wc.lpszClassName     = "OpenGL";                         // Set The Class Name

	RegisterClass(&wc);

	if (g_Fullscreen)                                              // Attempt Fullscreen Mode?
	{
		DEVMODE dmScreenSettings;                       // Device Mode
		memset(&dmScreenSettings, 0, sizeof(dmScreenSettings));// Makes Sure Memory's Cleared
		dmScreenSettings.dmSize = sizeof(dmScreenSettings); // Size Of The Devmode Structure
		dmScreenSettings.dmPelsWidth    = g_WinWidth;   // Selected Screen Width
		dmScreenSettings.dmPelsHeight   = g_WinHeight;  // Selected Screen Height
		dmScreenSettings.dmBitsPerPel   = bits;         // Selected Bits Per Pixel
		dmScreenSettings.dmFields = DM_BITSPERPEL|DM_PELSWIDTH|DM_PELSHEIGHT;

		// Try To Set Selected Mode And Get Results.  NOTE: CDS_FULLSCREEN Gets Rid Of Start Bar.
		if (ChangeDisplaySettings(&dmScreenSettings,CDS_FULLSCREEN)!=DISP_CHANGE_SUCCESSFUL)
		{
			// If The Mode Fails, Offer Two Options.  Quit Or Use Windowed Mode.
			if (MessageBox(NULL,"The Requested Fullscreen Mode Is Not Supported By\nYour Video Card. Use
Windowed Mode Instead?","NeHe GL",MB_YESNO|MB_ICONEXCLAMATION)==IDYES)
			{
			g_Fullscreen=FALSE;// Windowed Mode Selected.  Fullscreen = FALSE
			}
			else
			{
			// Pop Up A Message Box Letting User Know The Program Is Closing.
			MessageBox(NULL,"Program Will Now Close.","ERROR",MB_OK|MB_ICONSTOP);
			return FALSE;                   // Return FALSE
			}
		}
	}

	if (g_Fullscreen)                                      // Are We Still In Fullscreen Mode?
	{
		dwExStyle=WS_EX_APPWINDOW;                      // Window Extended Style
		dwStyle=WS_POPUP;                               // Windows Style
		ShowCursor(FALSE);                          // Hide Mouse Pointer
	}
	else
	{
		dwExStyle=WS_EX_APPWINDOW | WS_EX_WINDOWEDGE;      // Window Extended Style
		dwStyle=WS_OVERLAPPEDWINDOW;                                    // Windows Style
	}

AdjustWindowRectEx(&WindowRect, dwStyle, FALSE, dwExStyle);    // Adjust Window To True Requested Size

	// Create The Window
	hWnd=CreateWindowEx(   dwExStyle,                   // Extended Style For The Window
				"OpenGL",                   // Class Name
				  title,                    // Window Title
				dwStyle |                   // Defined Window Style
			WS_CLIPSIBLINGS |                   // Required Window Style
			WS_CLIPCHILDREN,                    // Required Window Style
```

```
                    0, 0,                    // Window Position
               WindowRect.right-WindowRect.left,              // Calculate Window Width
               WindowRect.bottom-WindowRect.top,             // Calculate Window Height
                    NULL,                    // No Parent Window
                    NULL,                    // No Menu
                  hInstance,                 // Instance
                   NULL);                   // Dont Pass Anything To WM_CREATE

     static     PIXELFORMATDESCRIPTOR pfd=        // pfd Tells Windows How We Want Things To Be
     {
          sizeof(PIXELFORMATDESCRIPTOR),       // Size Of This Pixel Format Descriptor
          1,                                  // Version Number
          PFD_DRAW_TO_WINDOW |                 // Format Must Support Window
          PFD_SUPPORT_OPENGL |                 // Format Must Support OpenGL
          PFD_DOUBLEBUFFER,                       // Must Support Double Buffering
          PFD_TYPE_RGBA,                       // Request An RGBA Format
          bits,                               // Select Our Colour Depth
          0, 0, 0, 0, 0, 0,                          // Colour Bits Ignored
          0,                                   // No Alpha Buffer
          0,                                  // Shift Bit Ignored
          0,                                  // No Accumulation Buffer
          0, 0, 0, 0,                              // Accumulation Bits Ignored
          16,                                  // 16Bit Z-Buffer (Depth Buffer)
          0,                                  // No Stencil Buffer
          0,                                  // No Auxiliary Buffer
          PFD_MAIN_PLANE,                      // Main Drawing Layer
          0,                                  // Reserved
          0, 0, 0                              // Layer Masks Ignored
     };

     hDC = GetDC(hWnd);

     PixelFormat=ChoosePixelFormat(hDC,&pfd);
     SetPixelFormat(hDC,PixelFormat,&pfd);
     hRC=wglCreateContext(hDC);
     wglMakeCurrent(hDC,hRC);

     ShowWindow(hWnd,SW_SHOW);
     SetForegroundWindow(hWnd);
     SetFocus(hWnd);
     ResizeGLScene(g_WinWidth, g_WinHeight);

     return TRUE;                                    // Success
}


/*************************************************************************/
LRESULT CALLBACK WndProc( HWND hWnd,           // Handle For This Window
               UINT uMsg,                     // Message For This Window
               WPARAM     wParam,             // Additional Message Information
               LPARAM     lParam)             // Additional Message Information
{
     switch (uMsg)                          // Check For Windows Messages
     {
          case WM_SYSCOMMAND:              // Intercept System Commands
               switch (wParam)             // Check System Calls
               {
                    case SC_SCREENSAVE:  // Screensaver Trying To Start?
                    case SC_MONITORPOWER:      // Monitor Trying To Enter Powersave?
                    return 0;                // Prevent From Happening
               }
               break;                       // Exit

          case WM_CLOSE:                   // Did We Receive A Close Message?
               PostQuitMessage(0);         // Send A Quit Message
               return 0;                   // Jump Back
```

```
          case WM_SIZE:                    // Resize The OpenGL Window
               g_WinWidth = LOWORD(lParam);
               g_WinHeight = HIWORD(lParam);
               ResizeGLScene(g_WinWidth, g_WinHeight);
               return 0;                      // Jump Back

          case WM_KEYDOWN:
               {
                    switch(wParam)
                    {
                         case VK_ESCAPE:
                              Quit(0);

                         case VK_F4:
                              g_ToggleFullscreen = true;
                              break;
                   case VK_F8:

                    if(Tsize_flag == 0 ){
                   Tsize_flag = 1;
                       break;
                    }
                    if(Tsize_flag == 1 ){
                       Tsize_flag = 2;
                       break;
                    }
                    if(Tsize_flag == 2 ){
                       Tsize_flag = 0;
                       break;
                    }
                    break;


                         default: break;
                    }
                    return 0;
               }
     }
     // Pass All Unhandled Messages To DefWindowProc
     return DefWindowProc(hWnd,uMsg,wParam,lParam);
}
/*********************************************************************/

int WINAPI WinMain(  HINSTANCE  hInstance,           // Instance
                         HINSTANCE  hPrevInstance,  // Previous Instance
                         LPSTR      lpCmdLine, // Command Line Parameters
                         int        nCmdShow)         // Window Show State
{
     MSG       msg;                              // Windows Message Structure
     BOOL  done=FALSE;                           // Bool Variable To Exit Loop

     // Create Our OpenGL Window
     if (!CreateGLWindow("Flexiter", 16))
     {
          return 0;                                          // Quit If Window Was Not Created
     }

     InitScene();

     while(!done)                                // Loop That Runs While done=FALSE
     {
          if (PeekMessage(&msg,NULL,0,0,PM_REMOVE))    // Is There A Message Waiting?
          {
               if (msg.message==WM_QUIT)         // Have We Received A Quit Message?
               {
```

```
                done=TRUE;                  // If So done=TRUE
            }
            else                        // If Not, Deal With Window Messages
            {
                TranslateMessage(&msg);         // Translate The Message
                DispatchMessage(&msg);          // Dispatch The Message
            }

            if (g_ToggleFullscreen)
            {
                g_Fullscreen = !g_Fullscreen;
                g_ToggleFullscreen = false;

                KillGLWindow();
                CreateGLWindow("Flexiter", 16);
                g_Engine->ResetHWND(hWnd);

            }

        }
        else                                // If There Are No Messages
        {
            DrawGLScene();                      // Draw The Scene
            SwapBuffers(hDC);                   // Swap Buffers (Double Buffering)
        }
    }

    // Shutdown
    KillGLWindow();                             // Kill The Window
    return (msg.wParam);                       // Exit The Program
}
```

B. Image Segmentation based on Active Contours Code

% Image Segmentation Based On Active Contours
% Author: Chen Wang
% Developed by using Matlab

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all;
close all;

[ filename, pathname ] = uigetfile( '*', 'Image' );

% if user cancelled dialog box then exit
if( filename == 0 )
   return;
end;


Img1 = imread( strcat( pathname, filename ));
Img11=double(Img1(:,:,1));

sigma=1.2;      % scale parameter in Gaussian kernel for smoothing.
G=fspecial('gaussian',15,sigma);
Img_smooth1=conv2(Img11,G,'same');    % smooth image by Gaussiin convolution

[Ix1,Iy1]=gradient(Img_smooth1);


f1=Ix1.^2+Iy1.^2;


g1=1./(1+f1);      % edge indicator function.
```

```
epsilon=1.5;      % the papramater in the definition of smoothed Dirac function       1.5

timestep=2.0;    % time step                                                                    5
mu=0.04;              % coefficient of the internal (penalizing) energy term P(\phi)         0.04
                                % Note: the product timestep*mu must be less than 0.25 for stability!

lambda=6.0;    % coefficient of the weighted length term L(\phi)       5
alf=-6.0;              % coefficient of the weighted area term A(\phi);       1.5
                                % Note: Choose smaller value for weak object bounday, such as the cell image in this demo.

                                % define initial level set function (LSF) as -c, 0, c at points outside, on
                                % the boundary, and inside of a region R, respectively.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

n_pixel=9;        % how many pixels u want to calculate

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

s_n_pixel=sqrt(n_pixel);
if mod(s_n_pixel-1,2)==0
      step_min=(s_n_pixel-1)/2;
      step_max=(s_n_pixel-1)/2;
else
      step_min=floor((s_n_pixel-1)/2);
      step_max=step_min+1;
end

R_f(4,3,3)=0;

figure;imagesc(Img1);hold on;

title('Several Specific Pixels Picking');

[xi,yi,vals] = impixel(Img1);

[a1,a2]=size(vals);


R_f=select(a1,n_pixel,xi,yi,double(Img1),step_min,step_max,R_f);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

 [nrow1, ncol1]=size(Img11);

 c=4;

 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 L=1;                              % half of length       1.5
 H=1;                              % half of height       1.5
 stepgap=4;                        % search distance      4
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

 u2=zeros(nrow1,ncol1);
 u0=c*ones(nrow1,ncol1);

% rc=10;           % range of pixel
 p1=0;             % min and max offset
 u2=search1(u2,u0,double(Img1),n_pixel,stepgap,R_f,nrow1,ncol1,L,H,c,p1);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 figure;imagesc(Img1);colourmap(grey);hold on;

 [c2,h2] = contour(u2,[0 0],'w');

title('Initial Contour');
pause(3.5);
% start level set evolution
for n=1:30      %    250
   u2=EVOLUTION_S (u2, g1 ,lambda, mu, alf, epsilon, timestep, 1);
```

```
        pause(0.05);

            imagesc(Img1);colourmap(grey);hold on;
            [c2,h2] = contour(u2,[0 0],'w');

            iterNum=[num2str(n), ' iterations'];
            title(['Low Resolution, ',iterNum]);
            hold off;

end
figure;imagesc(Img1);colourmap(grey);hold on;
[c2,h2] = contour(u2,[0 0],'w');
  hold off;

totalIterNum=[num2str(n), ' iterations'];
title(['Final Contour, ', totalIterNum]);
```

C. Shape Recognition in Satellite Image Code

%Shape Recognition in Satellite Image.
%Author: Chen Wang
% Developed by using Matlab

```
close all
clear all
ImageFile1='building_p32_b.bmp';
% Step 1: Read image Read in
RGB = imread(ImageFile1);
figure,
imshow(RGB),
title('Original Image');
BW=RGB;
[B,L] = bwboundaries(BW, 'noholes');

% Step 2: Determine objects properties
STATS = regionprops(L, 'all');
curve = fopen('houseCurve5.mel', 'wt');
h1=0;
a = 1; b = 6;
h2 = a + (b-a) * rand(1,1);
Csize=size(B);
for j=1:Csize(1)
h2 = a + (b-a) * rand(1,1);
Csize1=size(B{j,1});
k1=Csize1(1);
if j==1
fprintf(curve,'{\n');
fprintf(curve,'int $k1;\n');
else
fprintf(curve,'\n');
end
house_no='int $h001=%d; \n';
house_no1='for ($k1=4;$k1<$h009;$k1++) \n';
s_k3=find(house_no=='1');
s_h1=find(house_no1=='9');
if(j<10)
house_no(s_k3(1))=int2str(j);
house_no1(s_h1(1))=int2str(j);
end
if(j>=10&&j<100)
house_no((s_k3(1)-1):(s_k3(1)))=int2str(j);
house_no1((s_h1(1)-1):(s_h1(1)))=int2str(j);
end
if(j>=100&&j<1000)
house_no((s_k3(1)-2):s_k3(1))=int2str(j);
house_no1((s_h1(1)-2):s_h1(1))=int2str(j);
end
fprintf(curve,house_no,k1);
chen1='matrix $House001_1[              ][3]=<<';
s_k2=find(chen1=='1');
if(j<10)
```

```
chen1(s_k2(1))=int2str(j);
end
if(j>=10&&j<100)
chen1((s_k2(1)-1):(s_k2(1)))=int2str(j);
end
if(j>=100&&j<1000)
chen1((s_k2(1)-2):s_k2(1))=int2str(j);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
s_k1=find(chen1=='[');
if(k1<10)
chen1(s_k1(1)+1)=int2str(k1);
end
if(k1>=10&&k1<100)
chen1((s_k1(1)+1):(s_k1(1)+2))=int2str(k1);
end
if(k1>=100&&k1<1000)
chen1((s_k1(1)+1):(s_k1(1)+3))=int2str(k1);
end
if(k1>=1000&&k1<10000)
chen1((s_k1(1)+1):(s_k1(1)+4))=int2str(k1);
end
fprintf(curve,chen1);
for i=1:k1
    if (i==k1)
    fprintf(curve,'%6.2f,%6.2f,%6.2f',B{j,1}(i,2),h1,B{j,1}(i,1));
    else
   fprintf(curve,'%6.2f,%6.2f,%6.2f;',B{j,1}(i,2),h1,B{j,1}(i,1));
    end
end
fprintf(curve,' >>;\n');
h2 = a + (b-a) * rand(1,1);
chen1='matrix $House001_2[              ][3]=<<';
s_k1=find(chen1=='[');
if(j<10)
chen1(s_k2(1))=int2str(j);
end
if(j>=10&&j<100)
chen1((s_k2(1)-1):(s_k2(1)))=int2str(j);
end
if(j>=100&&j<1000)
chen1((s_k2(1)-2):s_k2(1))=int2str(j);
end
if(k1<10)
chen1(s_k1(1)+1)=int2str(k1);
end
if(k1>=10&&k1<100)
chen1((s_k1(1)+1):(s_k1(1)+2))=int2str(k1);
end
if(k1>=100&&k1<1000)
chen1((s_k1(1)+1):(s_k1(1)+3))=int2str(k1);
end
if(k1>=1000&&k1<10000)
chen1((s_k1(1)+1):(s_k1(1)+4))=int2str(k1);
end
fprintf(curve,chen1);
  for i=1:k1
    if (i==k1)
    fprintf(curve,'%6.2f,%6.2f,%6.2f',B{j,1}(i,2),h2,B{j,1}(i,1));
    else
   fprintf(curve,'%6.2f,%6.2f,%6.2f;',B{j,1}(i,2),h2,B{j,1}(i,1));
    end
end
fprintf(curve,' >>;\n');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%
curve_1='curve -p $House009_1[0][0] $House009_1[0][1] $House009_1[0][2] -p $House009_1[1][0] $House009_1[1][1]
$House009_1[1][2] -p $House009_1[2][0] $House009_1[2][1] $House009_1[2][2] -p $House009_1[3][0] $House009_1[3][1]
$House009_1[3][2]; \n';
s_k4=find(curve_1=='9');
size_k4=size(s_k4);

for p=1:size_k4(2)
if(j<10)
```

```
curve_1(s_k4(p))=int2str(j);
end
if(j>=10&&j<100)
curve_1((s_k4(p)-1):(s_k4(p)))=int2str(j);
end
if(j>=100&&j<1000)
curve_1((s_k4(p)-2):s_k4(p))=int2str(j);
end
end
fprintf(curve,curve_1);
fprintf(curve,house_no1);
curve_12='{ curve -a -p $House009_1[$k1][0] $House009_1[$k1][1] $House009_1[$k1][2] curve1;} \n';
s_k5=find(curve_12=='9');
size_k5=size(s_k5);
for p=1:size_k5(2)
if(j<10)
curve_12(s_k5(p))=int2str(j);
end
if(j>=10&&j<100)
curve_12((s_k5(p)-1):(s_k5(p)))=int2str(j);
end
if(j>=100&&j<1000)
curve_12((s_k5(p)-2):s_k5(p))=int2str(j);
end
end
fprintf(curve,curve_12);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
curve_2='curve -p $House009_2[0][0] $House009_2[0][1] $House009_2[0][2] -p $House009_2[1][0] $House009_2[1][1]
$House009_2[1][2] -p $House009_2[2][0] $House009_2[2][1] $House009_2[2][2] -p $House009_2[3][0] $House009_2[3][1]
$House009_2[3][2]; \n';

s_k6=find(curve_2=='9');
size_k6=size(s_k6);
for p=1:size_k6(2)
if(j<10)
curve_2(s_k6(p))=int2str(j);
end
if(j>=10&&j<100)
curve_2((s_k6(p)-1):(s_k6(p)))=int2str(j);
end
if(j>=100&&j<1000)
curve_2((s_k6(p)-2):s_k6(p))=int2str(j);
end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(curve,curve_2);
fprintf(curve,house_no1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
curve_22='{ curve -a -p $House009_2[$k1][0] $House009_2[$k1][1] $House009_2[$k1][2] curve2;} ';
s_k7=find(curve_22=='9');
size_k7=size(s_k7);
for p=1:size_k7(2)
if(j<10)
curve_22(s_k7(p))=int2str(j);
end
if(j>=10&&j<100)
curve_22((s_k7(p)-1):(s_k7(p)))=int2str(j);
end
if(j>=100&&j<1000)
curve_22((s_k7(p)-2):s_k7(p))=int2str(j);
end
end
fprintf(curve,curve_22);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(curve,' $s_x=-$HSizeX/2;\n');
fprintf(curve,' $s_z=-$HSizeZ/2; \n');
fprintf(curve,' move $s_x 0 $s_z curve1;    \n');
fprintf(curve,' move $s_x 0 $s_z curve2;    \n');
fprintf(curve,'loft -ch 1 -u 1 -c 0 -ar 1 -d 1 -ss 2 -rn 0 -po 1 -rsn true "curve2" "curve1";    \n');
fprintf(curve,' planarSrf curve1 ; \n');
fprintf(curve,' planarSrf curve2 ; \n');
fprintf(curve,' delete curve1 curve2; \n');
end
fprintf(curve,' }\n');
```

```
fclose(curve);
```

D. 3D Environment Reconstruction and Flood Simulation Code

// 3D Urban Environment Reconstruction and 3D Flood Simulation
// Author : Chen Wang
// Developed by using Maya Embed Language

```
global string $image_h1="./image/SwanseaH.bmp" ;    // terrain heightmap
global string $image_T1="./image/SwanseaT.jpg";      // terrain texturemap
global string $image_house1="./image/brick_01.jpg";    // house texturemap      house2.jpg
global string $image_w2="./image/water2.bmp";       // flood      texturemap2 for transparency
global string $image_river="./image/water1.jpg.jpg";          // river      texturemap
global string $image_forest="./image/leaf/forest1.jpg";          //      forest texturemap grassbam.jpg    forest1.jpg
global string $image_sea="./image/water3.bmp";                  //      sea        texture
global string $image_c1="./image/f1.bmp"; // load house region by segmented image
global string $image_house2="./image/brick_02.jpg";    // house texturemap      house2.jpg
global string $image_house3="./image/brick_03.jpg";    // house texturemap      house3.jpg
global string $model_house1="./model/house1.mb";        // house model
global string $model_forest1="./model/mesh_4.mb";        // forest model

global string $image_bumpforest="./image/heightmap_forest3.bmp";    // forest bump image,

global float $HSizeX,$HSizeZ; //heightmap Width and Height size
global int $scale_HY=50;    // scale terrain
global int $u1=2;                // terrain subdivision in u
global int $v1=2;                // terrain subdivision in v
global float $p_t=0.4;    // parameter for transparency 0.6529
global int $density_house, $density_tree,$density_tree1;
global string $Nfile,$Cfile,$Cfile1,$Cfile2;
global matrix $s[20][5];
$density_house=4;
//house 255 255 0

$s[0][0]=225.0/255.0;$s[0][1]=255.0/255.0;
$s[1][0]=225.0/255.0;$s[1][1]=255.0/255.0;
$s[2][0]=0.0/255.0; $s[2][1]=50.0/255.0;

//forest 0 255 0
$density_forest=4;

$s[3][0]=0.0/255.0;$s[3][1]=10.0/255.0;
$s[4][0]=220.0/255.0;$s[4][1]=255.0/255.0;
$s[5][0]=0.0/255.0; $s[5][1]=10.0/255.0;

source "./curveToPoly.mel";      //        proc curveToPoly()

proc floodsimulation()

{

global string $s_terrain;
global string $s_forest;
global string $s_river;
global string $s_sea;
global string $s_house;
global string $s_flood;
global string $s_bumpf;
global string $s_smoothhill;

$s_terrain=stringArrayToString({"source ","\"./reconstruction_terrain.mel\""}, "");

$s_forest=stringArrayToString({"source ","\"./reconstruction_forest_mesh.mel\""}, "");

$s_river=stringArrayToString({"source ","\"./reconstruction_river.mel\""}, "");

$s_sea=stringArrayToString({"source ","\"./reconstruction_sea.mel\""}, "");

$s_house=stringArrayToString({"source ","\"./reconstruction_house.mel\""}, "");
```

```
$s_flood=stringArrayToString({"source ","\"./simulation_floodplane.mel\""}, "");
$s_bumpf=stringArrayToString({"source ","\"./image_bumpforest.mel\""}, "");
$s_smoothhill=stringArrayToString({"source ","\"./smooth_hill.mel\""}, "");
string $window = `window -title "FloodSimulation"    -widthHeight 150 500 `;
       columnLayout -columnAttach "both" 12 -rowSpacing 8 -columnWidth 250;
       button -label "Renew" -command "file -f -new";
       button -label "3D Real World" -command $s_terrain;
       button -label "Forest Reconstruction" -command $s_forest;
       button -label "River Reconstruction" -command $s_river;
       button -label "Sea Reconstruction" -command $s_sea;
       button -label "Urban Reconstruction" -command $s_house;
       button -label "Flood Simulation" -command $s_flood;
       button -label "Bump Imagef" -command $s_bumpf;
       button -label "Smooth Hill" -command $s_smoothhill;
       button -label "Close" -command ("deleteUI    -window " + $window);
    showWindow;
}
floodsimulation();
```

E. List of Publications

[1]. C. Wang, T. R. Wan and I. J. Palmer, "Large-Scale 3D Environmental Modelling and Visualisation for Flood Hazard Warning", Submitted to IEEE Transactions on Geoscience and Remote Sensing.

[2]. C. Wang, T. R. Wan and I. J. Palmer, "3D Large-Scale and Detailed Environment Reconstruction from Active Contours Based Segmentation and Intelligent Shape Match Method", Submitted to International Journal of Computers & Graphics, Elsevier.

[3]. C. Wang, T. R. Wan and I. J. Palmer, "Urban Flood Risk Analysis for Determining Optimal Flood Protection Levels Based on Digital Terrain Model and Flood Spreading Model", Submitted to International Journal of Visual Computer, Springer.

[4]. C. Wang, T. R. Wan and I. J. Palmer, "An Adaptive Multi-Seed Geometric Active Contour Model for River Recognition", IEEE International Conference on Audio, Language and Image Processing 2008(ICALIP2008), Shanghai, China, 7-9 July 2008. ISBN:9781424417230

[5]. C. Wang, T. R. Wan and I. J. Palmer, "3D Topological Reconstruction of Large-Scale Urban Environment from Satellite Images and Real Elevation Data", Proceedings of the Ninth Informatics Workshop, pp. 222-225, Bradford, UK, June 2008. ISBN:9781851432516

[6]. C. Wang, T. R. Wan and I. J. Palmer , "A Real-time Dynamic Simulation Scheme for Large-Scale Flood Hazard Using 3D Real World Data", IEEE 11th International Conference Information Visualisation, pp. 607-612, IEEE Computer Society Press, Zurich, Switzerland, 4 - 6 July 2007. ISBN: 9780769529004

[7]. C. Wang, T. R. Wan and I. J. Palmer, "Robust Segmentation for Satellite Images Based on Active Contours", Proceedings of the Eighth Informatics Workshop, pp. 222-225, Bradford, UK, June 2007. ISBN:9781851432462

[8]. C. Wang and T. R. Wan, "3D Dynamic Simulation for Natural Disaster Prevention", Proceedings of the Seventh Informatics Workshop, pp. 178-181, Bradford, UK, Mar 2006. ISBN: 1851432329

[9]. C. Wang and T. R. Wan, "Modelling and Reconstructing Real World Using Ground-Based and Airborne Measurement Data", IEEE/IEE/EPSRC Postgraduate Research Conference in Electronics, Photonics, Communications & Networks, and Computing Science (PREP2005), pp. 285-286, Lancaster, UK, March 30 – April 1, 2005.

References

Abidi. M.A and Gonzalez. R.C, Eds. Data Fusion in Robotics and Machine Intelligences Academic Press, San Diego, 1992.

Abramovitz. J. N, 2001.Unnatural Disasters. Worldwatch Paper 158, ISBN: 1-878071-60-2.

Adams. R and Bischof. L, "Seeded region growing," IEEE Trans. Pattern Anal. Machine Intell., vol. 16, pp. 641–647, 1994.

Ahuja. N and Nash. C,"Octree representation of moving objects",Computer Vision, Graphics, and Image Processing, 26:207–216, 1984.

Airault, S., Jamet, O., Leymarie, F., 1996. From manual to automatic stereo plotting: evaluation of different road network capture process, IAPRS, vol. 31, pp. 14-18.

Alemseged. Tamiru. Haile and Rientjes.T.H.M, "Effects of LIDAR DEM Resolution in flood modelling:A model sensitivity study for the city of Tegucigalpa, Honduras",ISPRS Workshop "Laser scanning 2005", Enschede, the Netherlands, September 12-14, 2005.

Alfredsen. K and Saether. B, "An object-oriented application framework for building water resource information and planning tools applied to the design of a flood analysis system", Journal of Environmental Modelling and Software, 15: 215-224, 2000.

Alfredsen. K, "Simulation of human impacts on flood regions–the design of an object-oriented integration framework", Proc of Hydroinformatics, University of Iowa, 2000.

Ameri, B., 2000. Feature based model verification: a new concept for hypothesis validation in building reconstruction, IAPRS, Amsterdam, vol. 33, part B3, pp. 24-35.

Amini. J, Lucas. C, Saradjian. M, Azizi. A and Sadeghian. S, Fuzzy Logic System for Road Identification Using IKONOS Images. Photogrammetric Record 17(99), pp. 493–503, 2002.

Andersen.Hans.Erik, Stephen E. Reutebuch, and Gerard F. Schreuder, "Bayesian object recognition for the analysis of complex forest scenes in airborne laser scanner data", In ISPRS Commission III, Symposium 2002 September 9 - 13, 2002.

Axelsson. Peter, "Processing of laser scanner data algorithms and applications",ISPRS Journal of Photogrammetry and Remote Sensing, 54(2-3):138–147, 1999.

Bagan H, Yang. Y and Ma. J, "Land cover classification from MODIS EVI times-series data using SOM neural network", Int J Remote Sens 26:4999–5012,2005.

Baltsavias, E.P., Mason, S., Stallmann, D., 1995. Use of DTMs/DSMs and orthoimages to support building extraction, Workshop on AEMOASI, Basel, pp. 199-210.

Baltsavias. E, S. Mason and D. Stallmann, Use of dtms/dsms and orthoimages to support building extraction, in Workshop on Automatic Extraction of Man-Made Objects from Aerial and Space Image (A. Gruen, O.Kuebler, and P. Agouris, Eds.), pp. 199–210, Basel, 1995.

Barbara. D. P, Joseph.V.L, McAnally. W. H and Thomas. W. A, "Users Guide to RMA2 WES Version 4.5", 2003. http://chl.wes.army.mil/software/tabs/docs.htp

Barre de St.-Venant, A.J.C., 1871. Théorie du mouvement non permanent des eaux avec application aux crues des rivie`res et a` l'introduction des mare´es dans leur lits. Compte rendu des seances de l'Academie des Sciences 73, 147–154 and 237–240.

Barzohar, M., Cooper, D.B., 1996. Automatic finding of main roads in aerial images by using geometric-stochastic models and estimate, IEEE T-PAMI, 18(7): 707-721.

Bates PD, Marks KJ, Horritt MS. Optimal use of high-resolution topographic data in flood inundation models. Hydrol Process 2003;17:537–57.

Bates, P.D., De Roo, A.P.J., 2000. A simple raster-based model for flood inundation simulation. Journal of Hydrology 236, 54–77.

Bates. P.D, Stewart. M.D, Siggers. G.B, Smith. C.N, Hervouet, J.-M and Sellin. R.J.H, Internal and external validation of a two-dimensional finite element code for river flood simulations. Proc of the Institute of Civil Engineers, Water, Maritime and Energy, 1998.

Baumgartner, A., Steger, C., Mayer, H., Eckstein, W., 1997. Multi-resolution, semantic objects and context for road extraction, Workshop on Semantic Modelling for the Acquisition of Topographic Information from Images and Maps, Basel, pp. 140-156.

Begnudelli L and Sanders BF. Conservative wetting and drying methodology for quadrilateral grid finite volume models. J Hydraul Eng 2006;133(3).

Begnudelli. L and Sanders BF. Unstructured grid finite volume algorithm for shallow-water flow and transport with wetting and drying. J Hydraul Eng 2006;132(4):371–84.

Boughton, W. and Droop, O., 2003. Continuous simulation for design flood estimation--a review.Environmental Modelling & Software, 18(4): 309-318.

Bradford SF and Sanders BF. Finite-volume model for shallow-water flooding of arbitrary topography. J Hydraul Eng 2002;128(3):289–98.

Bradford SF and Sanders BF. Performance of high-resolution, non-level bed, shallow-water models. J Eng Mech 2005;131(10):1073–81.

Braun. C, T. H. Kolbe, F. Lang,W. Schickler, V. Steinhage, A. B. Cremers,W. Forstner, and L. Plumer, Models for photogrammetric building reconstruction, Comput. Graphics 19, 1995, 109–118.

Brenner, C., 2000. Towards fully automatic generation of city models, IAPRS, 17-22 July, Amsterdam, vol. 33, part B3, pp. 85-92.

Brenner.C, 1999, Interactive modelling tools for 3D building reconstruction,Photogrammetric Week, Heidelberg, pp. 23-34.

Brouwers Lisa and Hansson Karin, "Scenario Simulations: Modelling of Flood Management Strategies", Elsevier Science, 2002.

Brouwers. L, “Spatial and Temporal Modelling of Flood Management Policies in the Upper Tisza Basin”, YSSP Report. Int. Inst. for Applied Systems Analysis (IIASA), Austria, 2002.

Brunn, A., Weidner, U., 1998. Hierarchical Bayesian nets for building extraction using dense digital surface models, ISPRS JPRS, 53: 296-307.

Busch, A., 1996. A Common Framework for the Extraction of Lines and Edges. In: International Archives of Photogrammetry and Remote Sensing, Vol. (31) B3/III, pp. 88–93.

Campbell. J. B, Introduction to Remote Sensing. The Guilford Press, New York, 1997.

Carter. Glenn. S and Shankar. Ude, “Creating rectangular bathymetry grids for environmental numerical modelling of gravel-bed rivers”, Applied Mathematical Modelling. Volume 20, No. 11, pp. 699-708, 1997.

Caselles.V, R. Kimmel, and G. Sapiro. Geodesic active contours. Int’l J. Comp. Vis., 22:61–79, 1997.

Chan. T and Vese. L, “Active contours without edges”, IEEE Trans. Imag. Proc, vol. 10, pp. 266-277, 2001.

Chang. Y. L and Li. X, “Adaptive image region growing”, IEEE Transactions on image Processing, 3: 868 – 872, 1994.

Chen X, Vierling L and Deering D, “A simple and effective radiometric correction method to improve landscape change detection across sensors and across time”, Remote Sens Environ 98:63–79, 2005.

Chen. H. H and Huang. T. S, “A survey of construction and manipulation of octrees”, Computer Vision, Graphics, and Image Processing, 43:409–431, 1988.

Chen. S. Y, Lin. W. C and Chen. C. T, “Split and merge image segmentation based on localized feature analysis and statistical tests”, CVGIP – Graphical Models and Image Processing, 53(5): 457 -475. 1991.

Cobby. D.M, Mason, D.C, Horritt. M.S, Bates, P.D, 2003, Two dimensional hydraulic flood modelling using a finite-element mesh decomposed according to vegetation and topographic features derived from airborne scanning laser altimetry. Hydrological Processes 17 (10).

Cohen WB, Maiersperger TK and Yang Z, “Comparisons of land cover and LAI estimates derived from ETM+ and MODIS for four sites in North America: a quality assessment of 2000/2001 provisional MODIS products”, Remote Sens Environ 88:233–55, 2003.

Collins. R, Y. Cheng, C. Jaynes, F. Stolle, and X.Wang, Task driven perceptual organization for extraction of rooftop polygons, in Proc on Computer Vision, pp. 888–893, 1995.

Coppin P, Jonckheere I and Nackaerts K, “Digital change detection methods in ecosystem monitoring: a review”, Int J Remote Sens 10:1565–96, 2004.

Couloigner, I., Ranchin, T., 2000. Mapping of urban areas: a multi-resolution modelling approach for semi-automatic extraction of streets, PE&RS, 66(7): 867-874.

Crowder. D and Diplas. P, “Using two-dimensional hydrodynamic models at the scales of ecological importance”, Journal of Hydrology, Volume 230, Issues 3-4, pp. 172-191,2000.

Cumani. A, edge detection in multispectral images, CVGIP: Computer Models and Image Processing, 53, 40-51, 1991.

Curran.P. J, Foody. G. M and Van Gardingen P. R. Scaling-up. In: van Gardingen, P. R., Foody, G. M. and Curran, P. J (Eds.) Scaling-up: From Cell to Landscape, Cambridge University Press. pp. 1-5, 1997.

Danish Hydraulic Institute, “MIKE 21 Reference Manual and User Guide”, 2001.

Dartmouth Flood Observatory http://www.dartmouth.edu/floods.

David.F, Rick. B and Jiang. J.H, “Three-dimensional computational flow modelling and high resolution flow surveys for fisheries environmental studies on the upper Columbia River”, Proceedings of Hydro Vision 2002 conference. Portland, Oregon, July 2002.

Davies. E, Machine Vision: Theory, Algorithms and Practicalities, Academic Press, Chap. 5, 1990.

Debevec. P, “Image-Based Techniques for Digitizing Environments and Artifacts,” Proc. 4th Int’l Conf. 3-D Digital Imaging and Modelling (3DIM 03), IEEE Press, pp.234-242, 2003.

DeFries R, Hansen M and Townshend J, “ Global discrimination of land cover from metrices derived from AVHRR data sets”, Remote Sens Environ 54:209–22,1995.

Demigny. D and Kamle. T, A discrete expression of Canny’s criteria for step edge detector. Performance evaluation, IEEE Trans on Pattern Analysis and Machine Intelligence, vol. 19, n 11, p. 1199-1211, November 1997.

Dial, G., Gibson, L. and Poulsen, R., 2001. IKONOS Satellite Imagery and Its Use in Automated Road Extraction. In: Automatic Extraction of Man-Made Objects from Aerial and Space Images (III), Balkema Publishers, Lisse, The Netherlands, pp. 357–367.

Domac. A and Suzen. ML, “Integration of environmental variables with satellite images in regional scale vegetation classification”, Int J Remote Sens 27:1329–50,2006.

Doucette, P., Agouris, P., Stefanidis, A. and Musavi, M., 2001. Self-Organized Clustering for Road Extraction in Classified Imagery. ISPRS Journal of Photogrammetry and Remote Sensing 55, pp. 347–358.

Du Y, Teillet PM and Cihlar J, “Radiometric normalization of multitemporal high-resolution satellite images with quality control for land cover change detection”, Remote Sens Environ 82:123–34,2002.

Duarte.J.P, “Customizing Mass Housing: A Discursive Grammar for Siza's Malagueira Houses”, Phd dissertation, Department of Architecture, Massachusetts Institute of Technology, Cambridge, MA,2002.

Dubicki. Alfred and Stronska. Kinga, "Development of a physical based Flood Forecasting System on the Upper and. Middle Odra", DHI- Water and Environment, Flood Management Department, Denmark and UK,2000.

Duchaineau. M. A, M. Wolinsky, D. E. Sigeti, M. C. Miller, C. Aldrich, and M. B. Mineev-Weinstein, "ROAMing terrain: real-time optimally adapting meshes", in IEEE Visualization, pp. 81–88, 1997.

Duda T and Canty M , " Unsupervised classification of satellite imagery: choosing a good algorithm", Int J Remote Sens 23:2193–12,2002.

Earth Observatory 1993
http://earthobservatory.nasa.gov/Newsroom/NewImages/images.php3?img_id=16881

EDOS, ftp://e0srp01u.ecs.nasa.gov/.

Elaksher, A.F., Bethel, J.S., 2002. Building extraction using LiDAR data, ACSM-ASPRS Annual Conference, 19-26 April, Washington DC, 9 p.

Elberink. S.O, Mass. H, 2000, The use of anisotropic height texture measures for the segmentation of airborne laser scanner data, IAPRS, Amsterdam, vol. 33, pp. 678-684.

Engman. E. T and Gurney. R. J, "Remote Sensing in Hydrology",Chapman and Hall, London,1991.

ERDAS Field Guide, 5th edition. ERDAS_, Inc., Atlanta, GA,1999.

Ermolieva. T, "The Design of Optimal Insurance Decisions in the Presence of Catastrophic Risks", Interim Report: IR-97-068. Int. Inst. for Applied Systems Analysis (IIASA), Laxenburg, Austria, 1997.

Estrela, T and Quintas. L, Use of GIS in the modelling of flows on floodplains. In: White, H.R., Watts, J. (Eds.), Second International Conference on River Flood Hydraulics, Wiley, Chichester, pp. 177–189,1994.

Evans E, Hall J, Penning-Roswell E, Sayers P, Thorne C, Watkinson A. Future flood risk management in the UK. Proc Inst Civ Eng – Water Management 2006;159(WM1):53–61.

Fan. J, J. Yu, G. Fujita, T. Onoye, L. Wu, and I. Shirakawa, Spatiotemporal segmentation for compact video representation, Signal Process. Image Commun. 16, 553–566, 2001.

FASW, http://www.gloucestershire.gov.uk/index.cfm?articleid=3328.

FD, http://www.thamesweb.com/topic.php?topic_name=Flood%20Defence.

Fischer.A, T. H. Kolbe, F. Lang, A. B. Cremers, W. Forstner, L. Plumer, and V. Steinhage, Extracting buildings from aerial images using hierarchical aggregation in 2d and 3d, Comput. Vision Image Understanding 72, 1998, 185–203.

Fischler, M.A., Heller, A.J., 1998. Automated techniques for road network modelling, DARPA Image Understanding Workshop, Monterey, CA, pp. 501-516.

Fischler, M.A., Tenenbaum, J.M., Wolf, H.C., 1981. Detection of roads and linear structures in low-resolution aerial imagery using a multisource knowledge integration technique, Computer Graphics & Image Processing, 15: 201-223.

Flemming. U, "More than the sum of its parts: the grammar of Queen Anne houses", Environment and Planning B: Planning and Design 14: 323-350, 1987.

French. J. R and Clifford. N. J, "Hydrodynamic modelling as a basis for explaining estuarine environmental dynamics: some computational and methodological issues", Hydrological Processes, Volume 14, Issues 11-12, pp. 2089-2108,2000.

Froehlich. D. C, "Finite element surface-water modelling system: two-dimensional flow in a horizontal plane", U. S. Department of Transportation Report No. FHWA-RD-92-057, 1992.

Ftacnik. M, Borovský. P and Samuelčík. M, "Low Cost High Quality 3D Virtual City Models",CORP 2004: 343-350,2004.

Gad S and Kusky T, " Lithological mapping in the Eastern Desert of Egypt, the Barramiya area, using Landsat thematic mapper (TM)", J Afr Earth Sci 44:196–202,2006.

Galland. J.C, Goutal. N and Hervouet. J.M, TELEMAC—a new numerical-model for solving shallow-water equations. Advances in Water Resources 14 (3), 138–148, 1991.

Gamba, P., Houshmand, B., 2002. Joint analysis of SAR, LIDAR and aerial imagery for simultaneous extraction of land cover, DTM and 3D shape of buildings, International Journal of Remote Sensing, 23(20): 4439-4450.

Garnesson, P., Giraudon, G., Montesinos, P., 1990. An image analysis system: application for aerial imagery interpretation, International Conference on Pattern Recognition, 16-21 June, Atlantic, vol. 1, pp. 210-212.

Geerken R, Zaitchik B and Evans JP, "Classifying rangeland vegetation type and coverage from NDVI time series using Fourier Filtered Cycle Similarity", Int J Remote Sens 26:5535–54,2005.

Geibel, R., Stilla, U., 2000. Segmentation of laser altimeter data for building reconstruction: different procedures and comparison, IAPRS, 17-22 July, Amsterdam, vol. 33, part B3, pp. 326-334.

Geman. S and Geman. D, Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. IEEE Transactions on Pattern Analysis and Machine Intelligence 6 (6), 721–741,1984

Gianinetto. Marco, Paolo Villa, and Giovanmaria Lechi, "Postflood Damage Evaluation Using Landsat TM and ETM+ Data Integrated With DEM", IEEE Transactions on Geoscience and Remote Sensing, Vol. 44, NO. 1, January 2006.

Gilvear. D. J, Bryant. R and Hardy.T, "Remote sensing of channel morphology and instream fluvial processes", Progress in Environmental Science. Volume 1, No. 3, pp. 257-284,1999.

Gomes Pereira, L.M., Wicherson, R.J., 1999. Suitability of laser data for deriving geographical information. A case study in the context of management of fluvial zones. ISPRS Journal of Photogrammetry and Remote Sensing 54, 105–114.

Gruen, A., Li, H., 1994. Semi-automatic road extraction by dynamic programming, IAPRS, pp. 324-332.

Gruen, A., Li, H., 1997. Linear feature extraction with 3-D LSB-Snakes, Workshop on AEMOASI, pp. 287-297.

Gruen. A and Navatia. R, Eds. Special Issue on Automatic Building Extraction from Aerial Images, Comput. Graphics Image Understanding 72(2), 1998.

Gruen. A, E. P. Baltsavias, and O. Henricsson, Eds. Automatic Extraction of Man-Made Objects from Aerial and Space Image (II), Birkh¨auser, Basel, 1997.

Guigues. L, Cocquerez. J.P and Le. Men. H, Scale-Sets Image Analysis. International Journal of Computer Vision 68 (3), 289–317, July,2006.

Haala, N., Anders, K.H., 1997. Acquisition of 3-D urban models by analysis of aerial images, digital surface models and existing 2-D building information, SPIE Conference on Integrating Photogrammetric Techniques with Scene Analysis and Machine Vision, Orlando, pp. 212-222.

Haala. N and Hanh. M, Data fusion for the detection and reconstruction of buildings, inWorkshop on Automatic Extraction of Man-Made Objects from Aerial and Space Image (I) (A. Gruen, O. Kuebler, and P. Agouris, Eds.), pp. 211–220, Birkh¨auser, Basel, 1995.

Haala. Norbert and Brenner. Claus, “Extraction of buildings and trees in urban environments”,ISPRS Journal of Photogrammetry and Remote Sensing, 54(2-3):130–137, 1999.

Haile Alemseged Tamiru and T.H.M.Rientjes, “Effects of LIDAR DEM Resolution in flood modelling:A model sensitivity study for the city of Tegucigalpa, Honduras”,ISPRS WG III/3, III/4, V/3 Workshop "Laser scanning 2005", Enschede, the Netherlands, September 12-14, 2005.

Hakim. S. F. El, “Semi-automatic 3D Reconstruction of Occluded and Unmarked Surfaces from Widely Separated Views,” Proc. ISPRS Commission V Symp, Ziti Publishing, pp. 143-148, 2002.

Hall. F.G, Strebel. D.E and Nickeson. J.E, “Radiometric rectification: toward a common radiometric response among multidate, multisensor images”, Remote Sens Environ 35:11–27, 1991.

Hanson. A. R and Riseman. E. M, “Segmentation of natural scenes”, Computer Vision Systems,New York: Academic Press, pp 129-163,1978.

Heckbert, P.S., Garland, M., 1997. Survey of polygonal surface simplification algorithms, Technical Report, Computer Science Department, Carnegie Mellon University, 29 p.

Hendrickx. M, J. Vandekerckhove, D. Frere, T. Moons, and L. Van Gool, 3d reconstruction of house roofs from multiple aerial images of urban areas, in IAPRS, pp. 88–95, 1997.

Henricson, O., Bignone, F., Willuhn, W., Ade, F., Kubler, O., Baltsavias, E., Mason, S., Gruen, A., 1996. Project Amobe: strategies, current status and future work, IAPRS, vol. 31, pp. 321-330.

Henricsson. O, The role of colour attributes and similarity grouping in 3d building reconstruction, Comput. Vision Image Understanding 72,163–184,1998.

Hervouet. J.M and Janin. J.M, 1994, Finite element algorithms for modelling flood propagation. In: Molinaro, P., Natale, L. (Eds.), Modelling flood propagation over initially dry areas, American Society of Civil Engineers, New York, pp. 243–256.

Hervouet. J.M and Van Haren. L, Recent advances in numerical methods for fluid flows. In: Anderson, M.G., Walling, D.E., Bates, P.D. (Eds.), Floodplain Processes, Wiley, Chichester, pp. 183–214,1996.

Hess. L. L, Melack. J. M and Simonett. D. S, "Radar detection of flooding beneath the forest canopy: a review", International Journal of Remote Sensing, 11, pp.1313-1325, 1990.

Hijjatoleslami. S. A and Kittler. J, "Region growing: A new approach," IEEE Trans. Image Processing, vol. 7, pp. 1079–1084, 1998.

Hirsch C, "Numerical computation of internal and external flows." 1990, Wiley J.

Hodges. B. R and Imberger. J, "Simple curvilinear method for numerical methods of open channels Journal of Hydraulic Engineering",Volume 127, No. 11, pp. 949-958,2001.

Hofmann, P., 2001. Detecting buildings and roads from IKONOS data using additional elevation information, GIS, 6: 28-33.

Hogg AJ and Pritchard D. The effects of hydraulic resistance on dambreak and other shallow inertial flows. J FluidMech 2004;501:179–212.

Hoppe, Smooth View-Dependant Level-of-Detail Control to Terrain Rendering, 1997.

Hoppe.H, "Smooth view-dependent level-of-detail control and its application to terrain rendering", IEEE Visualization 1998 Conference, 35-42.

Horritt, M.S. and Bates, P.D., 2002. Evaluation of 1D and 2D numerical models for predicting river flood inundation. Journal of Hydrology, 268(1-4): 87-99.

Horritt.M.S, "A methodology for the validation of uncertain flood inundation models", Journal of Hydrology, IEE, 2005.

Horwitz. S. L and Pavlidis. T, "Picture segmentation by a tree traversal. algorithm," J. ACM, vol. 23, pp. 368–388, Apr. 1976.

House of Commons, "Climate Change, Water Security and Flooding", Environment, Food and Rural Affairs Committee, Sixteenth Report of Session, 2004.

Hsieh. Y. C, D. M. McKeown, and F. P. Perlant. Performance evaluation of scene registration and stereo matching for cartographic feature extraction. IEEE Pattern Anal. Mach Intell. 14, 1992, 214–238.

Hu, X., Tao, V., 2003b. Automatic extraction of main-road centerlines from high resolution satellite imagery based on perceptual grouping, ASPRS Annual Conference, 3-9 May, Anchorage, AK, 9 p.

Hu, Y., Tao, V., 2002. Bald DEM generation and building extraction using range and reflectance LiDAR data, ACSM-ASPRS Annual Conference and FIG XXII International Congress, 19-26 April, Washington DC, 11 p.

Hu, Y., Tao, V., 2003. Automatic extraction of digital terrain models and road networks using multiple returns LiDAR data, ASPRS Annual Conference, 3-9 May, Anchorage, AK, 11 p. Alharthy, A., Bethel, J., 2003. Automated road extraction from LiDAR data, ASPRS Annual Conference, 3-9 May, Anchorage, AK, 8 p.

Hu. Y., Tao, V., Collins, M., 2003. Automatic extraction of buildings and generation of 3-D city models from airborne LiDAR data, ASPRS Annual Conference, Anchorage, AK, 12 p.

Hug, C., 1997. Extracting artificial surface objects from airborne laser scanner data,Workshop on AEMOASI, Basel, pp. 203-212.

IFPFC, http://www.environment-agency.gov.uk/homeandleisure/floods/38399.aspx

Jang J-D, Bartholome´ E, Viau AA, "Neural network application for cloud detection in SPOT VEGETATION images", Int J Remote Sens 27:719–36, 2006.

Jennings. A. A, "Modelling sedimentation and scour in small urban lakes", Environmental Modelling & Software, Volume 18, Issue 3, pp. 281-291,2003.

Jet Propulsion Laboratory, 2004. Shuttle Radar Topography Mission.

Jonathan, "Terrain Rendering at High Levels of Detail", 2000.

Kapur. J. N., P. K. Sahoo, and A. K. C. Wong.,"A new method for grey-level picture threshold using the entropy of the histogram", comput. Vision Graphics Image Process. 29, 273-285,1985.

Kass. M, Witkin. A and Terzopoulos. D, "Snakes: active contour models", Int'l J. Comp. Vis., vol. 1, pp. 321-331, 1987.

Katopodes ND and Strelkoff T. Computing two-dimensional dam-break flood waves. J Hydraul Div ASCE 1978;104(HY9):1269–88.

Kimia. B. B, Tannenbaum. A and Zucker S, "Shapes,shocks, and deformations I: the components of two-dimensional shape and the reaction-diffusion space", Int'l J.Comp. Vis., vol. 15, pp. 189-224, 1995.

Klang, D., 1998. Automatic Detection of Changes in Road Databases Using Satellite Imagery. In: International Archives of Photogrammetry and Remote Sensing, Vol. (32) 4/1, pp. 293–298.

Koning. H and Eizenberg. J, "The language of the prairie: Frank Lloyd Wright's prairie houses",Environment and Planning B: Planning and Design 8: 295-323,1981.

Koshimura, S, 2004. The December 26, 2004 Andaman-Nicobar Islands Earthquake Tsunami.

Lang. F and Forstner. W, 3d-city modelling with a digital one-eye-stereo system, in IAPRS, pp. 261–266, 1996.

Langley SK, Cheshire HM and Humes KS, "A comparison of single date and multitemporal satellite image classifications in a semi-arid grassland", J Arid Environ 49:401–11, 2001.

Laptev, I., 1997. Road Extraction Based on Line Extraction and Snakes [Mater Thesis], Royal Institute of Technology, Stockholm, Sweden, 71 p.

Leclerc. Y, Constructing simple stable descriptions for image partitioning. International Journal of Computer Vision 3 (1), 73–102,1989.

Lemmens, M., Deijkers, H., Looman, P., 1997. Building detection by fusing airborne laser altimeter DEMs and 2-D digital maps, IAPRS, vol. 32, pp. 42-49.

Lenka C and Milan C, "Supervised classification of plant communities with artificial neural networks", Journal of Vegetation Science 16:407–414,2005.

Lenney MP, Woodcock CE and Collins JB, "The status of agricultural lands in Egypt: the use of multitemporal NDVI features derived from LandsatTM", Remote Sens Environ 56:8–20,1996.

Li. C, Xu. C, Gui. C and Fox. M. D, Level set evolution without re-initialization: A new variational formulation. In IEEE Conference on Computer Vision and Pattern Recognotion (CVPR), volume 1, pages 430–436, 2005.

Li. J.B, Zhu X, Zhou GH, Cai BH. The compound effect of disastrous floods in Dongting Lake on concurrence of ecological disasters. Acta Ecol Sin2002;22:334–40.

Lillesand. T.M and Kiefer. R.W, Remote Sensing and Image Interpretation. John Wiley and Sons Inc., New York, 1994.

Lin B, Wicks JM, Falconer RA, Adams K. Integration of 1D and 2D hydrodynamic models for flood simulation. Proc Inst Civ Eng – Water Management 2006;159(WM1):19–26.

Lin. C. L, Q. Zheng, R. Chellappa, L. S. Davis, and X. Zhang, Site model supported monitoring of aerial images, in CVPR'94, pp. 694–699, 1994.

Lindstrom Peter and Pascucci Valerio, "Terrain Simplification Simplified: A General Framework for View-Dependent Out-of-Core Visualization", 2002.

Lindstrom Peter and Pascucci Valerio, "Terrain Simplification Simplified: A General Framework for View-Dependent Out-of-Core Visualization", IEEE Transactions on Visualization and Computer Graphics, 8(3), July-September 2002, 239-254.

Lindstrom Peter, David Koller, William Ribarsky, Larry F. Hodges and Nick Faust, "Real-Time, Continuous Level of Detail Rendering of Height Fields", Proceedings of ACM SIGGRAPH 96, August 1996, pp. 109-118.

Liow Y T. and Pavlidis. T, Use of shadows for extraction of buildings is aerial images, Comput. Vision Graphics Image Process 49, 1990, 242–277.

Liow. Y. T, "A contour tracing algorithm that preserves common boundaries between regions", CVGIP – Image Understanding, 53(3): 313 – 321, 1991.

Liu Yaolin and Miao Zuohua, "Key Technique of Floods Prevention Information System Based on GIS" Geoscience and Remote Sensing Symposium, IGARSS'05, 2005 IEEE International Volume 6, 25-29, Page(s):4395-4398, 2005.

Lowry. R. T, Langham. E. J. and Mudry. N, "A preliminary analysis of SAR mapping of the Manitoba flood", Satellite Hydrology, American Water Resources Association, Minneapolis, pp. 316-323, 1981.

Maas, H., Vosselman, G., 1999. Two algorithms for extracting building models from raw altimetry data, ISPRS JPRS, 54: 153-63.

Malladi.R, Sethian. J. A and Vemuri. B. C, "Shape modelling with front propagation: a level set approach", IEEE Trans. Patt. Anal. Mach. Intell., vol. 17, pp. 158-175, 1995.

Mark. O, Weesakul. S, Apirumanekul. C, Aroonnet. S.B and Djordjevic. S, Potential and limitations of 1D modelling of urban flooding. Journal of Hydrology, 284-299, 2004.

Marks K and Bates P, Integration of high-resolution topographic data with floodplain flow models. Hydrol Process 2000;14: 2109–22.

Masaya. Tanigawa, Tomoichi. Takahashi, Tesuhiko. Koto, Ikuo. Takeuchi and Itsuki.Noda, "Urban Flood Simulation as a component of integrated earthquake Disaster Simulation System", Proceedings of the 2005 IEEE, International Workshop on Safety, Security and Rescue Robotics Kobe, Japan, June 2005.

Mason, D.C., Cobby, D.M., Horritt, M.S., Bates, P.D., 2003. Floodplain friction parameterization in two-dimensional river flood models using vegetation heights derived from airborne scanning laser altimetry. Hydrological Processes 17 (9), 1711–1732.

Mayer, H., Laptev, I., Baumgartner, A., Steger, C., 1997. Automatic road extraction based on multi-scale modelling, context and snakes, IAPRS, vol. 32, part 3-2W3, pp.106-113.

Mayer. H, "Automatic Object Extraction from Aerial Imagery: A Survey Focusing on Buildings," Computer Vision and Image Understanding, vol. 74, no. 2, May 1999, pp. 138-149.

McKeown, D.M., Denlinger, J.L., 1988. Cooperative methods for road tracing in aerial imagery, CVPR, 5-9 June, Ann Arbor, pp. 662-672.

Mena. J and Malpica. J, 2003. Colour Image Segmentation Using the Dempster-Shafer Theory of Evidence for the Fusion of Texture, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. (34) 3/W8, pp. 139–144.

Milan Sonka, Vaclav Hlavac and Roger Boyle, "Image Processing Analysis and Machine Vision", Second Edition, TA1637.S66, 1998.

Mohammadzadeh, A., Tavakoli, A. and Zoej, M., 2004. Automatic Linear Feature Extraction of Iranian Roads from High Resolution Multi-Spectral Satellite Imagery,The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. (35) B3, pp. 764–767.

Morris MW. CADAM concerted action on dambreak modelling. Report SR 571,HR Wallingford; 2000.

Morten. Runge and Wium. Olesen. Kim, "Combined one- and two-dimensional flood modelling", Fourth Iranian Hydraulic Conference, 21-23 October, Shiraz, Iran, 2003.

Mumford. D and Shah. J, Optimal approximations by piecewise smooth functions and associated variational problems. Communications on Pure and Applied Mathematics 17 (4), 577–685,1989.

Munich. Re, 1998, Annual Review of Natural Catastrophes, 1998.

Neelz, S and Pender, G,The influence of errors in digital terrain models on flood flow routes. In: Ferreira, R.M.L., Alves, E.C.T.L., Leal, J.G.A.B., Cardoso, A.H. (Eds.), Proceedings of the International Conference on Fluvial Hydraulics. RiverFlow 2006, pp. 1955–1962.

Nevatia. J. Li. R and Noronha. S, User assisted modelling of buildings from aerial images, in CVPR'99, pp. 274–279, 1999.

Noronha. S and Nevatia. R, Detection and description of buildings from multiple aerial images, in IEEE Conf. on Computer Vision and Pattern Recognition, San Juan, Puerto Rico, pp. 588–594, 1997.

Oliver. M. A and Webster. R, "Kriging: a method of interpolation for geographical information system", INT. J. Geographical Information Systems, VOL. 4, No. 3, 1990.

Osher. S and Fedkiw. R, Level Set Methods and Dynamic Implicit Surfaces, Springer-Verlag, New York, 2002.

Otsu. N, A threshold selection method from grey-level histogram", IEEE Trans. Systems Man Cybernet. SMC-8, pp 62-66, 1978.

Paragios. N and Deriche. R, Geodesic active regions and level set methods for supervised texture segmentation. Int'l J. Comp. Vis., 46:223–247, 2002.

Park, J.S., Saleh, R.A., 2001. A comprehensive survey of extracting techniques of linear features from remote sensing imagery, Meeting on Road Centerline Extraction & Maintenance, 6-7 August, Santa Barbara, URL: http://www.ncgia.ucsb.edu/ncrst/.

Peteri, R. and Ranchin, T., 2003. Multiresolution Snakes for Urban Road Extraction from Ikonos and Qickbird. In: EARSeL Symposium.

Pohl. C and Genderen. J.L, Multisensor image fusion in remote sensing concepts methods and application,International Journal of Remote Sensing 19, 5 pp. 823-854,1998.

Prewitt. J. M. S and Mendelsohn M. L, The analysis of cell images, in Ann. New York Acad. Sci., Vol. 128, pp.1035-1053, New York Acad. Sci., New York, 1996.

Price, K., 1999. Road grid extraction and verification, IAPRS, vol. 32, pp. 101-106. Hinz, S., Baumgartner, A., 2000. Road extraction in urban areas supported by context objects, IAPRS, vol. 33, part B3, pp. 405-412.

Pw, http://webgis.wr.usgs.gov/pigwad/tutorials/planetarygis/planetary_gis_faq.htm#mirror

Rossignac. J and Borrel. P, "Multi-resolution 3D approximation for rendering complex scenes," Proc of Geometric Modelling in Computer Graphics, pp 453-465, June 1993

Rottensteiner, D.F., 2001. Semi-automatic extraction of buildings based on hybrid adjustment using 3D surface models and management of building data in a TIS [Ph.D. dissertation], Vienna University of Technology. URL: http://www.ipf.tuwien.ac.at/fr/.

Röttger,"Real-Time Generation of Continuous Levels of Detail for Height Fields", 1998.

Roux. M and McKeown. D, Feature matching for building extractiton from multiple views, in IEEE Conf. on Computer Vision and Pattern Recognition, pp. 46–53, 1994.

Ruskone, R., 1996. Road Network Extraction by Local Context Interpretation: Application to the Production of Cartographic Data [Ph.D. thesis], Marne-La-Vallee University, France.

Sagi Filin, "Surface clustering from airborne laser scanning data sagi filin",In ISPRS Commission III, Symposium 2002 September 9 - 13, 2002, Graz, Austria, 2002.

Sahar. L and Krupnik. A, Semiautomatic extraction of building outlines from large-scale aerial images, Photogram. Eng. Remote Sensing 65, 1999, 459–465.

Schalkoff. R.J, "Digital Image Processing and Computer Vision", New York, USA, 1989.

Scholze. S, Moons. T and Van.Gool. L, "A generic 3D model for automated building roof reconstruction", Proceedings of the ISPRS Commission V Symposium 34: 204-209, 2002.

Schroeder TA, Canty MJ and Yang Z, "Radiometric correction of multi-temporal Landsat data for characterization of early successional forest patterns in western Oregon", Remote Sens Environ 103:16–26, 2006.

Sequeira. V et al, “Hybrid 3D Reconstruction and Image-Based Rendering Techniques for Reality Modelling,” Proc.Int’l Conf. Videometrics and Optical Methods for 3D Shape Measurement, SPIE, vol. 4309, pp. 126- 136,2001.

Sester. M, 2000. Generalization based on least squares adjustment, IAPRS, 17-22 July, Amsterdam, vol. 33, part B4, pp. 931-938.

Shewchuk JR. Triangle: engineering a 2D quality mesh generator and Delaunay triangulator. In: Lin MC, Manocha D, editors. Applied computational geometry: towards geometric engineering. Lecture Notes in Computer Science, vol. 1148. Springer; 1996. p. 203–22. Software may be obtained at http://www-2.cs.cmu.edu/quake/triangle.html.

Shrestha.D.P and Zinck J.A, “Land use classification in mountainous areas: integration of image processing, digital elevation data and field knowledge (application to Nepal)”, Int J Appl Earth Obs and Geoinf 3:78–85,2001.

Shufelt. J. A, Exploiting photogrammetric methods for building extraction in aerial images, in IAPRS, pp. 74–79, 1996.

Shum. H. Y and Kang. S.B, “A Review of Image-Based Rendering Techniques,” Proc. Int’l Conf. Visual Comm. and Image Processing (VCIP 00), SPIE, vol. 4067, pp. 2-13, 2000.

Shyu M and Leou J , “A geneticle algorithm approach to colour image enhancement”, Pattern Recognition 31:871–880,1998.

Sippel. S. J, Hamilton. S. K, Melack. J. M and Choudery. B. J, “Determination of inundation area in the Amazon River floodplain using the SMMR 37 GHz polarisation difference”, Remote Sens. Environ. 48, 1994, pp.70-76.

Sithole. G and Vosselman. G, 2004. Experimental comparison of filter algorithms for bare-earth extraction from airborne laser scanning point clouds. ISPRS Journal of Photogrammetry and Remote Sensing 59, 85–101.

Smith. L. C, “Satellite remote sensing of river inundation area, stage, and discharge: a review”, Hydrological Processes, 11, pp. 1427-1439, 1997.

Sobek, 2004. Sobek: managing your flow, Delft, Netherlands.

Sohn Y and Qi J , “Mapping detailed biotic communities in the Upper San Pedro Valley of southeastern Arizona using landsat 7 ETM + data and supervised spectral angle classifier”, Photogramm Eng Remote Sens 71:709–18,2005.

Sohn Y and Rebello NS, “Supervised and unsupervised spectral angle classifiers”, Photogramm Eng Remote Sens 68:1271–80, 2002.

Song C, Woodcock CE and Seto KC, “Classification and change detection using Landsat TM data: when and how to correct atmospheric effects?”, Remote Sens Environ 75:230–44,2001.

Song Jeong Heon, Soo Hee Han, Ki Yun Yu, and Yong Il Kim, "A study on using LiDAR intensity data for land cover classification", In ISPRS Commission III, Symposium 2002 September 9 -13, 2002, Graz, Austria, 2002.

Song, J.H., Han, S.H., Yu, K., Kim, Y., 2002. Assessing the possibility of land-cover classification using LiDAR intensity data, IAPRS, 9-13 September, Graz, vol. 34, 4p.

Song. C and Woodcock. C.E , "Monitoring forest succession with multitemporal Landsat images: factors of uncertainty", IEEE Trans Geosci Remote Sens ,41:2557–67,2003.

SRF, http://sertit.u-strasbg.fr/documents/romania_2005/romania05_en.html

Steger, C., 1998. An Unbiased Extractor of Curvilinear Structures. IEEE Transactions on Pattern Analysis and Machine Intelligence 20, pp. 113–125.

Steger, C., Glock, C., Eckstein, W., Mayer, H., Radig, B., 1995. Model-based road extraction from images, Workshop on AEMOASI, Basel, pp. 275-284.

Stilla. U, Hajdu. A, 1994, Map-aided structural analysis of aerial images, IAPRS,pp. 475-482.

Stiny, G. and Gips, J, "Shape Grammars and the Generative Specification of Painting and Sculpture", In C.V. Freiman (ed.), Proceedings of IFIP Congress 71: 1460-1465. Republished in O.R. Petrocelli (ed.), The Best Computer Papers of 1971: 125-135, 1971.

Stiny. G and Mitchell. W.J, The Palladian grammar. Environment and Planning B: Planning and Design 5: 5-18, 1978.

Stoica, R., Descombes, X. and Zerubia, J., 2004. A Gibbs Point Process for Road Extraction from Remotely Sensed Images. International Journal of Computer Vision 57(2), pp. 121–136.

Stuart N, Barratt T and Place C, "Classifying the neotropical savannas of Belize using remote sensing and ground survey",J Biogeogr 33:476–90,2006.

Takase. Y, Sho. N, Sone. A and Shimiya. K, "Automatic generation of 3d city models and related applications," International Archives of Photogrammetry, Remote Sensing and Spatial Information Sci-ences, Vol. XXXIV-5/W10,2003

Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest, "Introduction to Algorithms", MIT Press, 1990.

Thomas. A. Funkhouser and Carlo.H.Sequin, "Adaptive display algorithm for interactive frame rates during visualization of complex virtual environments," Computer Graphics (SIGGRAPH93 Proceedings), volume 27, pp 247-254, August 1993.

Tomoharu Hori and Michiharu Shiiba, "Micro Model Simulation Tools for Performance-based Design of a Flood Risk Management System", Journal of Natural Disaster Science, Volume 26, Number 2, pp73-80, 2004.

Toro EF. Shock-capturing methods for free-surface shallow flows. Chichester, UK, 2001.

Townsend. P. A and Walsh. S. J, "Modelling floodplain inundation using an integrated GIS with radar and optical remote sensing", Geomorphology, 21, pp. 295-312, 1998.

Trias-Sanz. R, A metric for evaluating and comparing hierarchical and multi-scale image segmentations, Proceedings of the International Geosciences and Remote Sensing Symposium, vol. 8. Seoul, South Korea, pp. 5647–5650, July 2005.

Trinder, J.C., Wang, Y., 1998. Knowledge based road interpretation in aerial images, IAPRS, vol. 32, pp. 635-640.

Trinder, J.C., Wang, Y., Sowmya, A., Palhang, M., 1997. Artificial intelligence in 3-D feature extraction, Workshop on AEMOASI, 5-9 May, Ascona, pp. 257-266.

Tsai. A, Yezzi. A and Willsky. A. S, Curve evolution implementation of the mumford-shah functional for image segmentation, denoising, interpolation, and magnification. IEEE Trans. Imag. Proc., 10:1169–1186, 2001.

Tso B and Olsen R, "Combining spectral and spatial information into hidden Markov models for unsupervised image classification", Int J Remote Sens 26:2113–33, 2005.

U.S.Army Corps of Engineers, "HEC-RAS: River Analysis System", Hydraulic Reference Manual, Version 3.1, November, Hydrologic Engineering Center, Davis, California, 2002.

Ulrich.T, "Continuous LOD Terrain Meshing Using Adaptive Quadtrees", Game Technology, 2000. http://www.gamasutra.com/features/20000228/ulrich_pfv.htm

USGS Geographic Data,<http://edc.usgs.gov/geodata/> (Jan. 20, 2003).

USGS, 2004. Tsunami and Earthquakes at the USGS.

USGS, Digital Elevation Model Data http://edc.usgs.gov/glis/hyper/guide/usgs_dem, 2003a

Vemuri. B and Chen. Y, "Joint image registration and segmentation", Geometric Level Set Methods in Imaging, Vision, and Graphics, Springer, pp. 251-269, 2003.

Vestri, C., Devernay, F., 2001. Using robust methods for automatic extraction of buildings, CVPR, vol. 1, 8-14 December, Kanai, Hawaii, pp. 133-138.

Vosselman, G., 1999. Building reconstruction using planar faces in very high-density data, IAPRS, 8-10 September, Munich, vol. 32, 6 p.

Vosselman, G., de Gunst, M., 1997. Updating road maps by contextual reasoning, Workshop on AEMOASI, 5-9 May, Ascona.

Vosselman, G., de Knecht, J., 1995. Road tracing by profile matching and Kalman filtering, Workshop on AEMOASI, Basel, pp. 265-274.

Vosselman, G., Dijkman, S., 2001. 3-D building model reconstruction from point clouds and ground plans, IAPRS, vol. 34, part 3/W4, pp. 37-44.

Wallace. S, Hatcher. M, Priestnall. G and Morton. R, 2001, Research Into a Framework for Automatic Linear Feature Identification and Extraction. In: Automatic Extraction of Man-Made Objects from Aerial and Space Images (III), Lisse, Niederlande.

Walton G and Morgan K , "Removing cloud cover from Landsat TM data using space shuttle imagery", Compass 74:133–5,1998.

Wang Q and Tenhunen J , " Vegetation mapping with multitemporal NDVI in North Eastern China Transect (NECT)", Int J Appl Earth Obs Geoinf 6:17–31,2004.

Wang, Y and Trinder, J., 2000. Road network extraction by hierarchical grouping, IAPRS, vol. 33, pp. 943-949.

Wang, Z and Schenk, T., 2000. Building extraction and reconstruction from LiDAR data, IAPRS, 17-22 July, Amsterdam, vol. 33, part B3, pp. 958-964.

Wang. X., J. Lim, R. T. Collins, and A. R. Hanson, Extracting surface textures and microstructures from multiple aerial images, in CVPR'97, pp. 301–306, 1997.

Weidner, U., 1995. Building extraction from digital elevation models, 56 p. URL: http://www.ipb.uni-bonn.de/ipb/lit/abstracts95/weidner95.building.html.

Weidner, U., Forstner, W., 1995. Towards automatic building extraction from high resolution digital elevation models, ISPRS JPRS, 50, 38-49.

Wickland. D. E, "Future Directions for Remote Sensing in Terrestrial Ecological Research", In: Asar, G. (Ed.) Theory and Applications of Optical Remote Sensing. John Wiley and Sons Ltd., Chichester, pp. 691-72, 1989.

Wiedemann, C., Heipke, C., Mayer, H. and Hinz, S., 1998. Automatic Extraction and Evaluation of Road Networks from MOMS- 2P Imagery. In: International Archives of Photogrammetry and Remote Sensing, Vol. (30) 3/1, pp. 285–291.

Wiedemann. C, 2002. Improvement of road crossing extraction and external evaluation of the extraction results, IAPRS, 9-13 September, Graz, vol. 34, part 3A/B, 4 p.

Wonka.P, Wimmer. M, Sillion. F and Ribarsky. W, Instant architecture. In Proceedings of ACM SIGGRAPH 2003 / ACM Transactions on Graphics 22(3): 669-677,2003.

Xu M, Watanachaturaporn P and Varshney PK, "Decision tree regression for soft classification of remote sensing data", Remote Sens Environ 97:322–36,2005.

Yamagata. Y and Yasuoka. Y, "Classification of wetland vegetation by texture analysis method using ERS-1 and JERS-1 images", Proc. IGARSS '93, Tokyo, 1993, pp.1614-1616.

Ying. Xinya, Wang. Sam S. Y and Khan. Abdul A, "Numerical Simulation of Flood Inundation Due To Dam and Levee Breach", World Water Congress 2003.

Yoon. T, Kim. T, Park. W, Kim.T.G, Building segmentation using active contour model, Joint ISPRS Workshop on Sensors and Mapping from Space, 27-30 September, Hannover, ,1999.

Zalubowski. J. Personal Communication, Boxley Materials, Blue Ridge, VA,2003.

Zhan, Q., Molenaar, M., Tempfli, K., 2002a. Building extraction from laser data by reasoning on image segments in elevation slices, IAPRS, vol. 34, pp. 305-308.

Zhan, Q., Molenaar, M., Tempfli, K., 2002b. Hierarchical image object-based structural analysis toward urban land use classification using high-resolution imagery and airborne LiDAR data, International Symposium on Remote sensing of Urban Areas, Istanbul, 8 p.

Zhang, C, Baltsavias, E and Gruen, A, 2001. Knowledge-based image analysis for 3D road.

Zhang, Q and Couloigner, I., 2004. A Wavelet Approach to Road Extraction from High Spatial Resolution Remotely-Sensed Imagery. GEOMATICA 58(1), pp. 33–39.

Zlotnick. A, Carnine. P.D, 1993, Finding road seeds in aerial images, CVGIP: Image Understanding, 57, 243-360.

Zoppou. C and S. Roberts, Catastrophic collapse of water supply reservoirs in urban areas, Journal of Hydraulic Engineering, American Society of Civil Engineering, pp.686–695, 1999.