AUTOMATIC RECONSTRUCTION OF 3D ENVIRONMENT USING REAL TERRAIN DATA AND SATELLITE IMAGES

CHEN WANG¹, TAO RUAN WAN² AND IAN J PALMER²
¹The James Hutton Institute
Aberdeen, UK
²School of Computing, Informatics, and Media
University of Bradford, UK

ABSTRACT—This paper presents a novel 3D reconstruction method for large-scale 3D environments. There are three core components of our work: dynamic terrain modelling, river and water region identification and modelling using an active contour model and primitive shape matching method. Real-time environment reconstruction is constructed using real measurement data of GIS, in terms of digital elevation data and satellite image data. A Nona Tree Space Partitions (NTSP) algorithm is proposed for dealing with very large data processing and visualisation. A new geometric active contours model is used to automatically segment interesting image areas such as water or flooded regions, forest region and residential region. A primitive shape matching method is proposed to detect the residential objects, such as buildings and houses. The experimental results demonstrate that our approach is a promising one, which is able to deal with large environment reconstruction effectively.

Key Words: Automatic, Environment Modelling and Reconstruction, Image Segmentation, Geometric Active Contours, Primitive Shape Matching.

1. INTRODUCTION

Many 3D environment reconstruction systems have been developed due to their wide and useful applications such as flood forecast, forest fire simulation and geographical information systems (GIS). Recently, it has become increasingly common to use image based reconstruction techniques on modeling terrain [1][2][3]. A standard approach to creating a 3D model is to build it from scratch using tools such as CAD software, creating 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 modeling 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 like computer generated rather than real. Several recent techniques [4] 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. The 3D environment reconstruction system consists of automatic building extraction and reconstruction [5][6][7], road extraction and reconstruction [8][9][10], vegetation extraction and reconstruction [11][12][13]. Such applications require detailed models that are still usually created manually. In this sense, we believe that the novel technique for automatic environmental object detection and reconstruction presented in this paper will be very helpful.
Our approach integrates several technologies including dynamic digital terrain modeling, a new active contours model and a primitive mapping method. In this paper a number of experiments are described and the results show this approach is promising and effective in dealing with large terrain data and satellite images for environment reconstruction.

2. DYNAMIC DIGITAL TERRAIN MODELLING

Our approach uses real data, such as the data provided by SRTM (Shuttle Radar Topography Mission) [14], instead of a grey scale map which is a critical component of large-scale geometrical environment generation. The focus of most related works has been on the development of algorithms for visualization of large scenes [15][16][17]. However, these methods are slow to create dynamic terrain. In order to overcome this problem, we propose the Nona Tree Space Partitions (NTSP) and pyramidal data arrangement structure for digital terrain modelling which has two main advantages:

(i) 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 nine terrain nodes for world visualisation for navigation, a whole world can be visualized as long as the necessary GIS data available.

(ii) An in-core process with a pyramidal data arrangement structure is efficient for processing high resolution terrain with detailed image and interpolated elevation data.

2.1 Nona Tree Space Partitions for Real time Terrain 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 [18][19], we propose the Nona tree space partitions (NTSP) algorithm in the current work. NTSP is useful for planar navigation, however, an octree is applied for spatial data refinement. The centre of nine local regions is 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. The details of the NTSP algorithm can be found [20][23].

Figure 1 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 2 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 $N_2$, $N_3$ and $N_8$. The nodes $N_1$, $N_2$ 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.
2.2 Experiment Results

Figure 3 shows the far-distance view of the Skye region in Scotland reconstructed by the method proposed in the current work. The generated terrain geometry shown in Figure 4 is overlaid by satellite image corresponding to the same geographic area represented as the input to the virtual earth model.

3. ACTIVE CONTOURS BASED IMAGE SEGMENTATION

Once a digital terrain has been generated, from the reconstruction application point of view, we then need to identify which area needs to reconstructed, using the adaptive image segmentation algorithm. For this purpose, we propose a novel multi-resolution vector-valued framework for image segmentation based on active contours for extracting interesting terrain regions, such as water, river regions and forests.

This section is structured as follows: part 1 introduces the active contours model development; part 2 talks about initialization of multiple snakes; part 3 presents integration of active contours model with elevation data from low to high resolution; part 4 gives some examples to proof our method.
3.1 River and Forest Identifications and Reconstruction

River and water regions can be identified by image segmentation to partition an image into a number of non-overlapping water and land regions based on image intensity, texture information and an active contours model.

Active contour-based image segmentation is proposed in the current work. The object contour is obtained by using an active contour model in the current research via a contour evolution process. Initial object boundaries are first set located inside the object regions. The boundaries are then pushed by boundary energy towards final boundaries through an iteration process to reach a minimum energy position.

Let a curve \( C(x, y) \) be a final water boundary, where \( X = X(S) \) and \( Y = Y(S) \), we define that \( \varphi_0 \{ \varphi(x, y) = \varphi_0(x, y), \varphi(x, y) \in \mathbb{R}^2 \} \) is the initial set of the moving boundary \( \varphi(x, y) \). The total energy driving the boundary curve can be described as

\[
E_\varphi = E_{\text{int}} + E_{\text{ext}}
\]  

The first term of Equation (1) is related to an internal energy, and can be expressed as

\[
E_{\text{int}} = \int_s (\alpha \cdot |\varphi(x, y)|^2 + \beta \cdot |\varphi(x, y)|^p)^2 ds = \int_s (\alpha \cdot \frac{d\varphi}{ds}^2 + \beta \cdot \frac{d^2 \varphi}{ds^2}^2) ds
\]  

\( \alpha \) and \( \beta \) are two constants to control the weights between two energy terms. The two terms of equation (2) are related to curve continuation and shape deformation. The second term of equation (1) is related to an external energy, there can be various ways to define it. Using image edge functional features of intensity and textures, we have

\[
E_{\text{ext}} = \gamma \int_s |\nabla \sigma G (x, y) * I(x, y)|^2 ds
\]  

where \( G_\sigma(x, y) \) is a Gaussian function with zero mean and standard deviation \( \sigma \), and * indicates a convolution operator, and \( \gamma \) is a weight constant to control and balance external energy contribution. Using variation calculus, we can obtain a flood boundary motion equation

\[
\alpha \cdot \varphi(x, y)''' - \beta \cdot \varphi(x, y)''' - \nabla E_{\text{ext}} = 0
\]  

The final shape of the region boundary reached will have a minimum total energy by an evolution process to find solution for Equation (4).

3.2 Multi-Resolution Feature Vector and Initialization of Active Contours

Once the active contours model is developed, initialization of multiple snakes is required. Our approach has the advantages of dealing with complicated satellite images and being able to initialize the required active contours automatically based on color and texture features for identifying the target objects with elevation data and texture information.

This algorithm is used for automatically producing zero level curves. The procedure can be described as shown in Figure 5.

As shown in Figure 5, for an image \( I(x, y) \) with the size of \( s_1 \times s_2 \), texture information containing the image grey levels, colors, and their deviation in the interesting object regions analyzed. If there are \( m \) selected points, for each point \( I(x, y) \), we take it as the center of a rectangular cell and give the proper length (\( l \)) and width (\( w \)) of that cell. We then calculate the interior pixel values of each cell in order to get \( P_m(\text{min}) \), \( P_m(\text{max}) \), \( P_m(\text{ave}) \), \( P_m(\text{sdev}) \), \( P_m(\text{gra}) \) for the minimum, maximum, average and standard deviation values with terrain image gradient. Then \( m \) cells are compared with each other for the purpose of acquiring appropriate image features:
In particular, these image features will be stored in a vector $V$, and initial active contours are then set accord to the values inside that vector. Figure 6 shows how to initialize the active contours:

i) Scanning the image $I$ with square cells $C$;
ii) To determine how many cells are satisfied with following criteria:

\[
P(\text{min})_{\text{min}} < C_m(\text{min}) < P(\text{min})_{\text{max}}
\]
\[
P(\text{max})_{\text{min}} < C_m(\text{max}) < P(\text{max})_{\text{max}}
\]
\[
P(\text{ave})_{\text{min}} < C_m(\text{ave}) < P(\text{ave})_{\text{max}}
\]
\[
P(\text{sdev})_{\text{min}} < C_m(\text{sdev}) < P(\text{sdev})_{\text{max}}
\]
\[
P(\text{tgra})_{\text{min}} < C_m(\text{tgra}) < P(\text{tgra})_{\text{max}}
\]

iii) Assign initial red square shaped active contours. We ignore yellow square shaped cells because they don’t meet the requirements of ii).

3.3 Multi-Resolution Segmentation with Elevation Data

As the active contour is defined and initialized, we want to integrate this model with elevation data for further image processing. This approach can be summarized in the following way (Figure 7): converting original image and elevation data into a grey scale image, setting different threshold values for the grey scale image in order to get the coarse areas of target object and initializing snakes in the coarse areas based on vector-valued scanning algorithm. Finally, our active contours model is used for curve evolution which will extract the accurate object boundary.
After defining the boundary at low resolution, the next step is to pass the extracted contour to a high resolution for optimization. The method is defined in formulation 20:

\[ U_h(a_1 \cdot x - i, a_1 \cdot y - j) = U_l(x, y) \]  

(20)

where \( a_1 \) is the scaling factor, \( U_l \) and \( U_h \) represent the contour at low and high resolution, and \( i \) and \( j \) are iterative parameters.

All of these initial snakes will move toward the river boundary. By using Ray et al [22] method, we can straightforward merging of multiple contours. If one snake is getting close to the other one, then they will be merged together to form a bigger snake and so on.

3.4 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 that further complicate the issue. Manual and semi-automatic tracking of images are still the common methods for obtaining the segmentations of complicated images. In our experiment, we select the city of Limerick for our research object. The low resolution image of that city is shown in Figure 8 with the size of 300*200. Figure 9 represents the initial snakes generated from the feature vector algorithm.
All of these initial snakes will move toward the river boundary. If one snake gets close to the other one, then they will be automatically merged together for a bigger snake and so on.

Figure 10. Final contours in low resolution

Figure 11. Final optimized contours in high resolution

The final snake in low resolution is shown in Figure 10 which is just after 60 iterations. Finally, the river contour will be passed to a high resolution for optimization.

Figure 11 shows the final optimized contour after 10 iterations which is consistent with the river boundary.

Further, we summarise the experimental results in terms of object recognition accuracy percentage as graph in Figure 12. As shown in the graph, the precision rates of river, building and forest identification achieve good performances among twenty test image slices.

Figure 12. Percentage accuracy of recognised objects
4. ENVIRONMENTAL OBJECTS DETECTION AND SHAPE MATCH METHOD

We propose a primitive-based shape matching method for identifying and extracting shapes of environmental objects such as buildings and houses, which analyses the binary maps and finds the closest matching types of primitives to model the environmental objects.

4.1 The Basic Types of Primitive

A typical regular shape environmental object is modelled as a cuboid, and the spatial distribution of the building objects is assumed not crowded, i.e., without serious shadowing and superposition. The cubic object is described by seven geometric parameters, \( C(\theta, X, Y, Z, L, H, D) \): 3D position, 3D size and orientation.

Figure 13 shows the projections of rectangular cuboids for primitive types of buildings. Figure 14 shows the complicated building shapes, which can be approximated by combinations of primitives shape types.

![Figure 13. Primitive types](image)

![Figure 14. Combinations of primitive types](image)

4.2 Primitive Shape Matching

The shape matching method is widely used through templates in the spatial domain. Template operations can be expressed as prototype matching and could be very useful as an elementary image filter. 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. Extracted buildings image needs to be normalized (i.e. whole image size =
Let $W_k(i,j) = [W^k]</m>_{ij}$, $i,j=1,2, \ldots, 10$ be dimensional vectors formed from the above templates and $I(i,j) = [I_{ij}]$, $i,j=1,2, \ldots, 10$ be a vector formed from an equivalent array of the pixels of the local sub-images. The inner product of $W_k(i,j)$ and $I(i,j)$ is defined by

$$I_kW_k(i,j) = \sum_i \sum_j W_{ki} \cdot I_{ij}$$

(21)

In terms of extracting a particular primitive, the primitive which is registered by the above procedure (i.e. the best of sixteen 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.

Assuming that the best match in the local area is determined. A primitive of the $m_{th}$ type is then registered as a candidate. For example, 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. We then calculate the means of the intensity values of pixels in these three regions. Let these be $F_p(x,y)$, $F_a(x,y)$ and $F_b(x,y)$. The candidate primitive is flagged as a primitive provided that the following conditions are met:

$$F_p(x,y) > F_a(x,y)$$

(22)

$$F_p(x,y) > F_b(x,y)$$

(23)

$$\sigma^2_L(x,y) > TL$$

(24)

where $F_p(x,y)$ is the average intensity value of the candidate primitive pixels in $S_p$, and $F_a(x,y)$ and $F_b(x,y)$ are the average intensity values of the pixels of the two neighbouring background regions: $S_a$ and $S_b$. The parameter $\sigma^2_L(x,y)$ is the local variance of the intensity values of pixels in the current area being examined, and $TL$ is the pre-assigned threshold value for the variance. If any one or more of the above conditions are not satisfied, the region or local area is flagged as having no primitive.

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

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

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

(25)

where, in the present application, $m=4$.

(ii) The local variance is then determined by:

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

(26)

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., $F_p(x,y) > F_c(x,y)$, $F_p(x,y) > F_d(x,y)$ ...).

Figures 15 and 16 show a typical example of building extraction in a residential area using our method.
Further, we summarize the experimental results in terms of objects shape match accuracy percentage as graph in Figure 17. As shown in the graph, the precision rates of river, building and forest identification achieve good performances among twenty test digital terrain slices.

5. 3D ENVIRONMENT RECONSTRUCTION CASE STUDIES

5.1 Large-scale Environment Reconstruction

It is difficult for large scale environments to be 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 18 shows the 3D real environment of Chepstow generated by our digital terrain modelling method.
Automatic object extraction has been applied to the texture map of Chepstow for different areas of segmentation. There are four identified areas listed below registered with distinct colour for sea, river, forest and residential regions which are shown in Figure 19.

We automatically reconstruct the environment based on these segmented regions, which is represented in Figure 20. Figure 21 displays the close distance view of reconstructed 3D environment of Chepstow including sea model and detailed house models with different locations, orientations and sizes.
5.2 Regional Environment Reconstruction

In this part, two experiments have been carried out for validating our proposed environmental object detection and intelligent shape match method.

The first experiment test was conducted to reconstruct one part of the Leeds area. We have successfully extracted boundaries of buildings from satellite image of Leeds as shown in Figures 22 and 23. Then through our intelligent shape match method, the 3D regional environment in one part of the Leeds area has been successfully reconstructed. Figure 24 and Figure 25 show separately the far-distance and close-distance 3D views of the reconstructed part of the Leeds area.

The next experiment was conducted to reconstruct one part of the Tewkesbury area. The texture map of local region of Tewkesbury was obtained from Google Earth in Figure 26. The whole environmental image has been segmented and identified as shown in Figure 27: 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.
The generated terrain geometry is overlaid by satellite image corresponding to the same geographic area which is presented in Figure 28. We also provide a close distance view of Local region of Tewkesbury reconstruction environment in Figure 29.

6. CONCLUSION

In this paper, a new approach is proposed for modeling and reconstructing large-scale 3D environments. The automatic 3D reconstruction system is developed which integrates three core elements: a NTSP algorithm handling large terrain data in real-time; a new active contours model for segmentation of terrain image for identifying water, river, and forest regions; a new primitive shape matching method for identifying and extracting buildings from the original images. The experiment results show that our method is promising, and able to deal with massive GIS terrain data, automatically segment targeted image areas and rebuild environmental objects efficiently.
REFERENCES


ABOUT THE AUTHORS

C. Wang is a Landscape and Visualisation Scientist at the James Hutton Institute. He received his BEng at Soochow University, and a PhD at University of Bradford. In 2010, He joined the Macaulay Institute, as a research scientist in Integrated Land Use Science. His research interests include 3D modelling of landscapes; urban environment modelling and reconstruction; character and traffic animation; 3D real time flood simulation.

T. R. Wan is a senior lecturer in the School of Computing, Informatics and Media at the University of Bradford. He received his BSc and MSc at Xian University of Engineering Science and Technology, and a PhD at the University of Leeds. His current research interests are mainly focused on computational models for 3D visualisation and simulation. This includes cloth modeling, crowd simulation, large-scale 3D environment modeling, flooding modeling, multi-agent system and traffic simulation.

I. J. Palmer is the Dean of the School of Computing, Informatics and Media at the University of Bradford. He has a BSc in Electronics from the University of Sussex. He obtained an MSc from Brunel University and a PhD in Computer Animation from Sussex. His research interests are computer animation and interactive virtual environments. This includes the use of behavioral approaches to animation to provide a richer and easier to use simulation environment and its application to both computer animation and computer games development.