Vis Comput (2016) 32:1293–1309
DOI 10.1007/s00371-016-1227-y

ORIGINAL ARTICLE

Interactive 3D content modeling for Digital Earth

Faramarz Samavati[1] · Adam Runions[1]

Published online: 19 April 2016

Abstract Digital Earth is a global reference model for integrating, processing and visualizing geospatial datasets. In this reference model, various data-types, including Digital Elevation Models (DEM) and imagery (orthophotos), are universally and openly available for the entire globe. However, 3D content such as detailed terrains with features, man-made structures, 3D water bodies and 3D vegetation are not commonly available in Digital Earth. In this paper, we present an interactive system for the rapid creation and integration of these types of 3D content to augment Digital Earth. The inputs to our system include available data sources, such as DEM and imagery information depicting landscapes and urban environments. The proposed system employs sketch-based and image-assisted tools to support interactive creation of textured 3D content. For adding terrain features visible in orthophotos, and also the basin of water bodies, we use a multiscale least square surface fitting to generate an adaptive triangular subdivision. For modeling forests and vegetation, we use image-based techniques and take advantage of visible regions and colors of forests in orthophotos. For 3D man-made structures, starting from a single photograph, we provide a simple image-assisted sketching tool to extract these objects, correct for perspective distortion and place them into desired locations.

Keywords Digital Earth · Computer graphics · Interactive modeling · Sketch-based modeling · Subdivision surfaces · Image-based modeling

✉ Faramarz Samavati
samavati@ucalgary.ca

[1] Department of Computer Science, University of Calgary, Calgary, AB, Canada

1 Introduction

With the recent technological advances in geospatial capturing technologies, there has been increasing interest in Digital Earth (DE) and its applications. DE provides a reference model for the integration, management, visualization (on the virtual earth) and processing of geospatial data [17]. This model efficiently integrates a vast amount of geo-located information such as Digital Elevation Models (DEM), satellite imagery, orthophotos and vector-based features (i.e. road systems) [4]. Recent technological advancements and innovations are allowing us to capture, analyze and visualize vast amounts of information about our planet. This information is typically geo-referenced and associated with a specific location on the Earth's surface. At present, DE software systems already incorporate many of these data-types and are advancing toward supporting 3D contents such as buildings, vegetation and other landscape and urban elements.

A problem arising in this context is the dynamic nature of our world. This creates a constant demand for the creation and editing of data within the context of a DE framework. Developing interactive tools that support rapid 3D content creation and manipulation for integration into this framework can help to alleviate these demands and complements automatic reconstruction techniques.

To enhance the availability and accuracy of 3D geospatial data, we should identify challenges associated with different types of geospatial data within the context of DE. Geospatial data can be broadly categorized into four groups: point-based and statistical datasets (e.g. oil wells, house prices, reported crimes), 2D (i.e. imagery, vector data), 2.5D (i.e. Digital Elevation Models) and 3D (e.g. buildings, bridges, vegetation, bodies of water). Imagery is the most commonly available source of information about the Earth. For example, orthophotos (aerial photograph with uniform scale) are

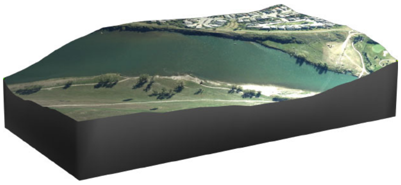

Fig. 1 A distorted river as a result of Imprecise DEM. DEM data are obtained from the *US Geological Survey*

available for many regions around the world [4]. Imagery provides raster information about landscapes and urban areas, but does not provide any 3D information. In contrast, Digital Elevation Models represent the rough geometry of the Earth's surface and incorporate salient features such as rivers, ridges and hills on a large scale. 3D geospatial data or *3D content* represents natural and man-made 3D objects on the Earth's surface [62].

Digital Elevation Models are available for the entirety of the Earth's surface. However, the quality and precision of DEM datasets depend on the acquisition techniques employed and vary drastically between datasets [34]. Several factors such as terrain roughness, sampling density, choice of interpolation algorithm, occluded terrain and vertical resolution affect the quality of DEMs [34]. Figure 1 depicts one of the typical issues arising in DEMs generated from low quality data. The characteristic geometry of important terrain features such as rivers, lakes, ridges and cliffs is not necessarily well represented by DEMs. Therefore, to improve the representation of these features, techniques for improving the accuracy and quality of DEMs are critical.

The 3D models (i.e. man-made structures, vegetation) required for detailed DE representations typically do not exist. Additionally, these objects are continually changing, and the input data required for 3D reconstruction are not always available. In recent years, various automatic methods have been proposed for reconstructing terrain and populating them with 3D content [40]. These methods have been used to reconstruct and visualize large portions of some cities [27,46]. Nevertheless, these methods have a number of limitations. To reconstruct a textured 3D object and compute its geographic coordinates, automatic methods typically require geo-referenced high quality input data as well as numerous photos of the object [40]. Moreover, objects have to be clearly visible and non-occluded in photos. In this regard, dense areas like forests and city centres are particularly difficult to reconstruct. Finally, automatic reconstruction methods cannot be used for scenarios where data are currently unavailable (i.e. urban planning, historical site reconstruction).

To address the issues discussed above, we propose a suite of sketch-based techniques. Sketch-based interfaces are a promising paradigm in interactive modeling, offering simple and natural ways in creating complex 3D shapes and perform other modeling tasks [41,43]. However, as observed by Schmidt et al. [49], drawing an accurate shape without assistance can be challenging. Using an image to guide the sketching process helps to create objects quickly and accurately [8,41,42]. In addition, the input image and the user sketch provide a model-image correspondence which is particularly useful within the context of our application scenario. Accordingly, in this paper, we introduce an interactive modeling system (Fig. 2) that uses available 2D imagery and DEMs to support the rapid creation of textured 3D content (e.g buildings, bridges, vegetation, bodies of water) and modification of the terrain geometry. Our system is designed to address content creation using an interactive semi-automatic approach. The final result of our system is similar to a 3D *maquette* or miniature model of a terrain including natural and man-made objects. For many regions, there are a large number of digital photographs that provide information which can be used in our system. Our proposed system can thus be used to enhance current data and create new 3D content. A variety of photos, orthophotos included, can be used as a guide for our system.

The idea of sketch-based 3D content creation for DE was introduced in our previous paper [25]. In this paper, we extend the initial approach of [25] by introducing a more comprehensive suite of interactive tools. In particular, the reconstruction tool for extracting 3D urban objects and interactive tool for integrating extracted models (Sect. 7) are new. Several new example results are also provided.

1.1 System overview

Figure 3 illustrates an overview of our system. Our system starts by specifying a region of interest (ROI) in the DE. A ROI is a rectangular area specified by the latitudes and longitudes of its corners, or alternatively using a cell index in the multiresolution reference model of the DE [16,38]. Our system retrieves the initial input data (e.g. DEMs and orthophotos) from the DE framework.

As depicted in Fig. 3, various landscape elements may appear in a given orthophoto. To support the creation and editing of landscape elements, we thus propose three sketch-based tools supporting the modeling of content based on the most common landscape elements appearing in orthophotos [53]: a terrain editor, as well as vegetation and body of water tools. The types of 3D content generated by these tools are illustrated in Fig. 3. The terrain editing tool (Sect. 3) facilitates interactive editing of DEM datasets to adjust the geometry for consistency with the features apparent in an orthophoto, such as rivers, roads and cliffs. The body of water tool (Sect. 4) interactively generates the volumetric geometry of a body of water. The vegetation tool (Sect. 5) interactively identifies and generates vegetation and plant ecosystems based on the photo's content.

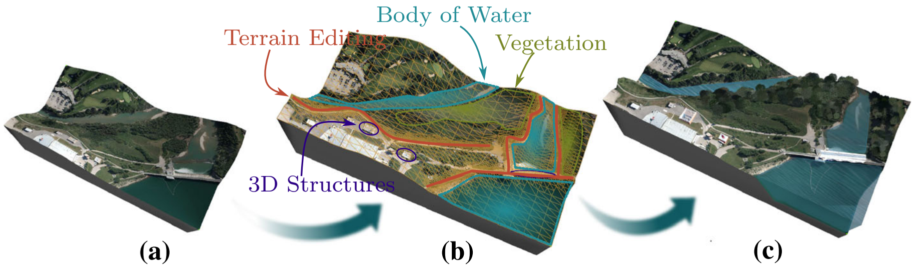

Fig. 2 3D Maquetter takes as input elevation data and an orthophoto (**a**). We employ a set of sketch-based tools (**b**) to create a 3D maquette of the region of interest (**c**)

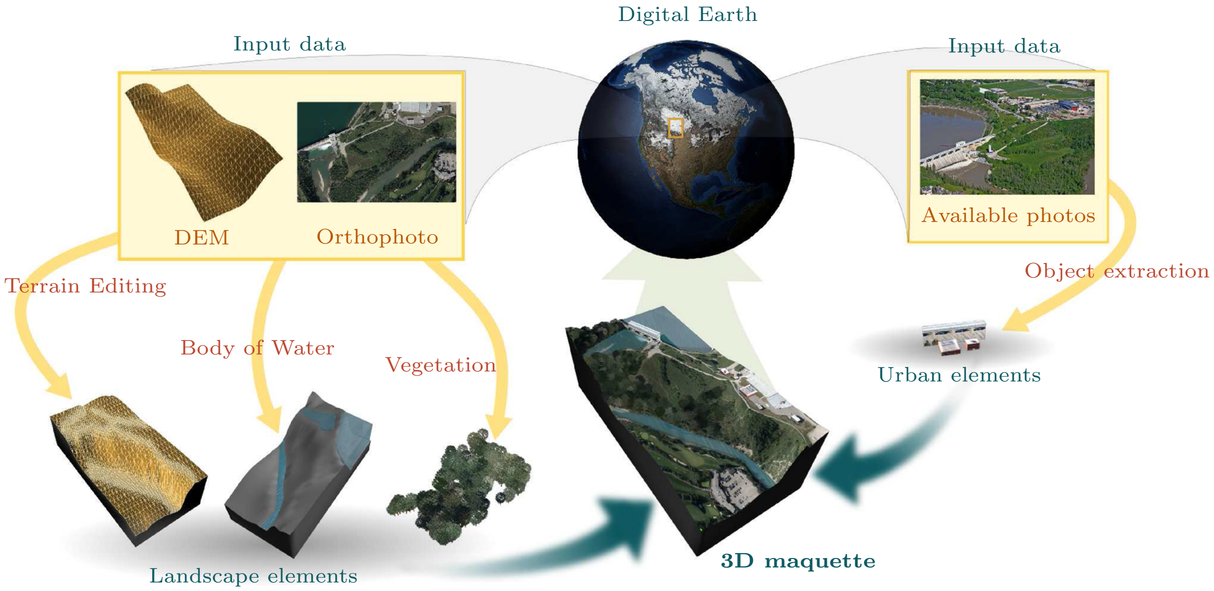

Fig. 3 System overview: the input data (the *yellow rectangles*) are retrieved from the DE. The orthophoto and DEM (*left*) are used for the creation of the landscape elements. Additionally, other available photos (*right*) are used for the extraction of man-made structures. The final result (3D maquette) is created and exported to DE

To enhance the availability of existing data, we propose extracting 3D man-made structures from available photos, and integrating them into a 3D maquette. Different types of photos can be used to extract 3D structures. Orthophotos as an input image provide the top of structures, which can potentially be used to create 3D models by extruding their footprints. Nevertheless, due to the view point of the camera, these photos have a number of limitations for reconstructing 3D structures. For example, the geometry of structures cannot be necessarily determined based on a single arial view (e.g. bridges and dams). To address this issue, any photo available online (Fig. 3) can also be loaded into our system to aid 3D object extraction. To support content creation for urban areas, we introduce the object extraction tool (Sect. 7) for creating textured 3D objects from a single photo. Using a single image as an input makes our tool effective at generating 3D content when limited input data are available [23,55]. As orthophotos are used extensively in our system to texture terrain and guide modeling, we present the clone tool for modifying and cleaning orthophotos (Sect. 6). Finally, the integrated result, the 3D maquette (consisting of a textured terrain together with all the created 3D models), is exported back to the DE (Fig. 3).

1.2 Contributions

Our main contribution is an image-guided sketch-based system for the rapid creation of 3D content and enhancement of existing content for a DE framework. In the proposed system, we have adapted a number of state-of-the-art techniques and modified them to address the challenges arising in the creation of 3D models for DE (as discussed in the preceding section). This leads to the following technical contributions:

- A sketch-based method and corresponding mathematical framework for modifying DEM datasets at multiple resolutions based on *features* visible in orthophotos.
- An image-based technique for modeling forests and tree stands based on an orthophoto.

- An interactive method for extracting 3D structures from a single photo and integrating them into a 3D maquette for export to DE.

2 Related work

The scale and resolution of geospatial datasets continually grow as data capture technologies improve. In this setting, data integration has emerged as one of the main challenges in leveraging these huge datasets. The Digital Earth framework has been proposed as an infrastructure to address this challenge [16].

The vision of a Digital Earth as "a digital replica of the entire planet" was first proposed in Al Gore's visionary talk on January 1998 [17]. Now, there are several frameworks built based on the concept of Digital Earth. Discrete Global Grid Systems (DGGSs) make such a representation possible by partitioning the Earth's surface into indexed cells (mostly regular) used to store the data associated with each index [16,38].

DE frameworks mostly accommodate a variety of 2D and 2.5D geospatial data formats and are advancing toward supporting 3D geospatial data. Geographical information systems (GIS) such as ESRI and BAE systems present various automatic and interactive tools for the creation and editing of geospatial data. These systems support interactive editing of DEM and 2D vector-based features (e.g. roads, bodies of water). However, 2D vector-based features are typically used to visualize various landforms and terrain features (e.g. rivers and roads) on the ground, and they do not have any 3D information. In contrast, the focus of our system is the sketch-based creation and editing of 2.5D and 3D geospatial data such as terrain, bodies of water, plants and urban areas to complement automatic reconstruction techniques. We discuss the modeling of the supported types of geospatial data in the summary of previous work below.

2.1 Terrain editing

As DEM datasets are often low resolution, terrain features are not necessarily represented accurately in the underlying DEM. This makes interactive image-based tools essential for the editing of DEM data to accurately represent terrain features, which must be accompanied by the introduction of details at multiple scales. Interactive terrain modeling and editing techniques have been the subject of extensive research. Fractal terrain deformation [54] and editing via control handles [22] were common aspects of earlier works. In contrast, direct manipulation methods which offer more natural interaction are increasingly preferred.

At present, interactive state-of-the-art techniques focus on: brush-based, exemplar-based and sketch-based interfaces. Brush-based methods [7] present the user with a set of interactive brushes for editing terrain. Although these brushes are well-suited to the sculpting of synthetic terrains, they do not support the editing of preexisting precise terrain features. Exemplar-based methods [6,61] edit terrain by finding the most similar region to a given area. However, terrain features often have unique characteristics and geometry which makes matching non-trivial and error prone.

The tool we propose is more closely related to sketch-based approaches. Sketch-based methods have been widely used for editing terrain and can be divided into two categories based on the viewpoint used to provide input. First person sketch-based systems [45,56] introduce interactive methods for editing terrain from a profile view. These methods provide limited control over the deformation of features.

Alternatively, interfaces also permit users to edit terrain from a number of different viewpoints. One such approach was presented by Gain et al. [14] who proposed a sketch-based technique for modeling synthetic terrain at a single resolution. However, precise editing of terrains based on features, such as rivers and cliffs with various slopes, was somewhat tedious and required multiple interactions. Bernhardt et al. [3] suggest a sketch-based method for deforming terrain based on features defined by elevation constraints. Their choice of constraint forced all features to have the same slope, in disagreement with real terrains. In contrast, our approach is a unique image-assisted sketch-based method which allows the terrain to be modified based on features obtained from orthophotos. Terrains can be edited freely from any point of view and the slopes are adjusted using a single stroke specifying the cross section of the terrain.

Attribute-based parametric modeling, such as the work of Gao and Rockwood [15], offers an alternative approach for terrain deformation using interpolating feature curves. For example, Staff et al. [33] employ this modeling approach to deform terrains using feature lines traced from 2D maps. This approach can potentially be used in image-assisted sketch-based systems. However, as we discuss in Sect. 3.2, to account for terrain features at different spatial scales it is beneficial to perform terrain deformations at multiple resolutions. Therefore, a mathematical model flexible enough to capture multiresolution surface deformations is needed. To do this, we phrase the mesh deformation as a hierarchical least-squares problem, using a subdivision scheme to relate the different levels of the hierarchy.

2.2 Vegetation

Plants are a ubiquitous part of urban areas and landscapes. Adding vegetation to DE frameworks increases their accuracy, as well as the realism of their visualization. Our system generates plant ecosystem based on an orthophoto using a sketch-based tool for specifying the areas covered with larger vegetation, such as shrubs and trees. In the literature, a number of methods have been proposed for generating trees and plant ecosystem which are either image-assisted or procedural.

Existing literature on generating trees using procedural modeling is vast. Photographs have also been used for modeling trees [35,55]. These methods are designed to model a tree from either a single image or multiple images. Our work is, however, most related to generating plants ecosystem. Simulation-based methods [11,28] have been extensively employed to generate forests and urban ecosystems. Hammes [19] proposed a technique for generating ecosystem based on DEMs. At the same time, the result of these methods may not be consistent with an orthophotos of a modeled region.

Although orthophotos are available for most regions of the earth's surface, their quality and viewpoint make them ineffective for reconstructing plant ecosystems automatically. Some methods have been proposed for counting trees in an orthophoto [48,60]. In contrast, we propose a sketch-based data-driven method for generating vegetation based on an orthophoto. Our method combines both procedural and image-based techniques to generate a plant ecosystem consistent with a given orthophoto. Distributing plants onto the terrain and coloring them based on an orthophoto is performed done similar to previously proposed procedural modeling and image-based techniques, respectively [28,55].

2.3 3D structures

Buildings, bridges and man-made 3D structures are an essential part of urban areas. A variety of methods have been proposed for extracting 3D objects from images [8,42]. Related works can be categorized based on input data: multiple photos, aerial photos and single image.

Extracting architecture from a sequence of images has been studied extensively, and it can be done either automatically [1,21,58] or interactively [10,18,52].

Generating wireframe and geometry based on aerial photos and texturing buildings using ground view images has been studied by Lee et al. [30–32]. Nonetheless, for applications such as urban site planning or historic site reconstruction, there may be inconsistencies between the structures depicted in different data sources. In addition, our method is capable of creating more varieties of 3D objects. Moreover, we facilitate simple sketching interaction for integrating results into 3D maquette to be exported into DE.

Many methods are focused toward extracting architecture from a single photo. Jiang et al. [23] present a novel method for modeling architecture from a single image. Their method is designed to model complex symmetric architectures.

Commercial modeling systems like *SketchUp* also support creating three-dimensional models on terrain. Users start by creating the geometry of structures by combining simple shapes, and texturing object using photos from different angles. This approach is powerful, but using this system requires basic 3D modeling skills.

3 Terrain editing tool

Orthophotos provide information about a variety of natural and man-made *features* such as rivers, cliffs, ridges and roads. Each of these features has unique characteristics that affect the geometry of the terrain (i.e. elevation, slopes, orientation). However, current DEM datasets are typically not sufficiently detailed to accurately capture these features. We introduce a sketch-based terrain editing tool to address this problem by identifying visually apparent features of the orthophotos. The geometry of features is defined by a *control curve*, where elevation along the curve, slopes and fields of influence are guided by the orthophoto (Fig. 4a). The orange curves in Fig. 4a illustrate an example of the left and right slopes around a feature specified by a *cross section curve*. The length of these strokes specifies the feature's field of influence. As depicted in Fig. 5, a variety of features can be represented by simply changing the form of these two curves.

Features generated using this method are typically more detailed than the highest resolution of existing DEM datasets. Therefore, to correct the geometry of features precisely in the terrain, the resolution around these features must be increased. Accordingly, we introduce a multiresolution terrain editing method. This method iteratively modifies the

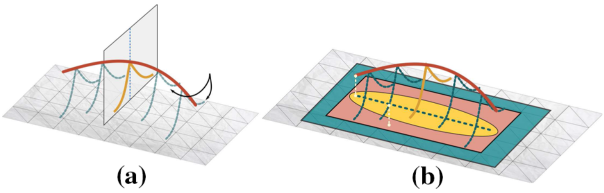

Fig. 4 Specification of a feature from a control curve. **a** The geometry of a feature is specified by the control (*red curve*) and cross section curve (*orange curve*). The cross section curve is placed at regular intervals along the control curve (*blue curves*). **b** The vertices within the *yellow* and *pink* regions are displaced to satisfy the positional constraints imposed based on the control and cross section curves, respectively. The energy minimization constraints are imposed on all the vertices within the *blue region*

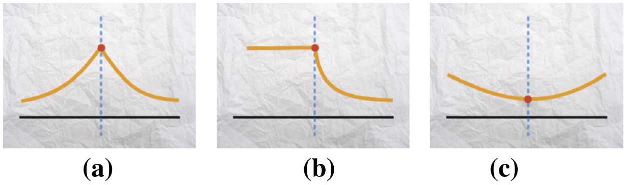

Fig. 5 Various terrain features can be represented by changing the slopes around the feature. **a** Ridge. **b** Cliff. **c** River bed

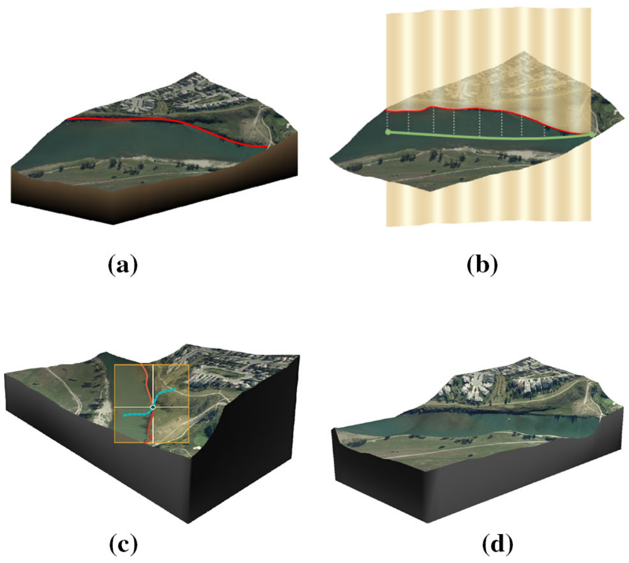

Fig. 6 Deformation tool applied to an example terrain. **a** Sketching the control curve (*red curve*) along the feature. **b** Specifying the elevation (*green curve*) along the control curve (*red curve*). **c** Specifying the slopes and fields of influence by sketching the cross section curve (*blue curve*). **d** The geometry of the river's edge is corrected based on the feature

terrain from low to high resolutions to fit a set of *positional* and *energy minimization* constraints. These constraints are created from the input strokes provided by the user. To increase computational efficiency, we adaptively subdivide the base terrain near features. The details of our method are provided in the remainder of this section, where Sect. 3.1 introduces our interface for sketch-based interaction, and Sect. 3.2 describes a deformation technique based on the input features this interface generates.

3.1 Sketch-based interaction

A feature's geometry is determined by three strokes which specify: a *control curve*, the elevation along the control curve and a *cross section curve* (Fig. 4a). First, the user sketches a control curve onto the terrain (Fig. 6a). The initial elevation of the control curve is then determined by the control curve's projection onto the terrain. To change the control curve's elevation, a *curtain* is automatically generated for sketching the elevation profile along the curve (Fig. 6b). To control the feature's slopes and fields of influence, the cross section curve (Fig. 4a) is sketched on two sides of the control curve (Fig. 6c). Finally, the terrain is deformed to best match the control and cross section curve (Fig. 6d).

3.2 Feature-based multiresolution terrain deformation

Digital Elevation Models are stored in two formats: *height map* and *triangular irregular network (TIN)* [34]. Due to its simplicity and computational efficiency, height maps have become the most prevalent format for representing DEM. In addition, preserving the regularity of the multiresolution terrain in the height map format is more challenging than TIN. Thus, our tool retrieves DEM from DE in the height map format. To capture the details of input features, we employ subdivision methods for increasing the resolution of the underlying DEM. To support both DEM formats, we use Loop subdivision [36], as suggested by [63], for this task by dividing each rectangular cell into two triangles. To export the modified DEM back into DE, several resolutions of DEM data must be stored in the height map format. To address this issue, we propose a hierarchical representation of the terrain resulted from the subdivision method. Therefore, given a terrain at the base resolution, we correct the geometry of the terrain at several resolutions, and we store each resolution in a height map which best fits the input features.

As described in Sect. 3.1, strokes are divided into two groups: control and cross section curves. Our goal is to deform the input terrain to best match control and cross section curves while preserving other characteristics of terrain. As the terrain is stored in a hierarchical representation, our algorithm has to support terrain deformation at different resolutions. To explain the terrain deformation technique based on input features, first we describe a method for approximating a single control curve, and then we present the terrain deformation technique that operates on multiple features.

3.2.1 Terrain deformation based on a single control curve

Given a terrain T with the base resolution T_0, we develop a multiresolution terrain deformation technique such that the terrain at resolutions $\{0, 1, \ldots, k\}$ best fits the given control curve. The control curve is defined by the polyline constructed from a set of 3D points denoted as $P = \{p_1, p_2, \ldots, p_m\}$ captured from the input stroke.

Various methods have been proposed for surface deformation [5]. Pusch and Samavati [47] introduce a technique which supports the local and multiresolution nature of our problem. They present a general framework for local constraint-based subdivision surface deformation. Starting from a given subdivision surface and a set of positional constraints, they solve a weighted least-squares problem to determine the control polygon of the subdivision surface. Although their method supports terrain deformation at multiple resolutions, it is unable to approximate a control curve more detailed than the initial terrain. Figure 7 illustrates an example of the terrain deformation based on the given control curve. Since the curve is more detailed than the initial terrain, displacing the original vertices is insufficient to accurately approximate the control curve at higher resolutions. Therefore, we extend their method to accurately approximate detailed control curves.

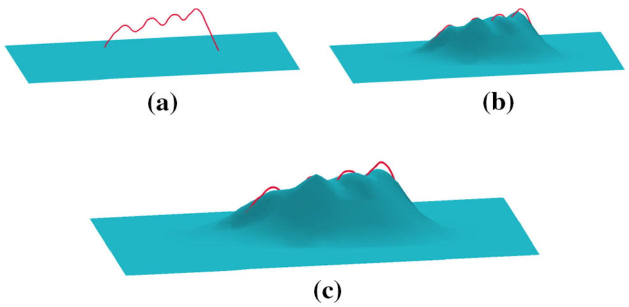

Fig. 7 The terrain deformation technique, proposed by [47], applied to multiple resolutions. As the input control curve is more detailed than the initial terrain, displacing the original vertices is not enough to accurately approximate the control curve at higher resolutions. **a** The input terrain and provided control curve. **b** The deformed terrain after one level of subdivision. **c** The deformed terrain after three levels of subdivision

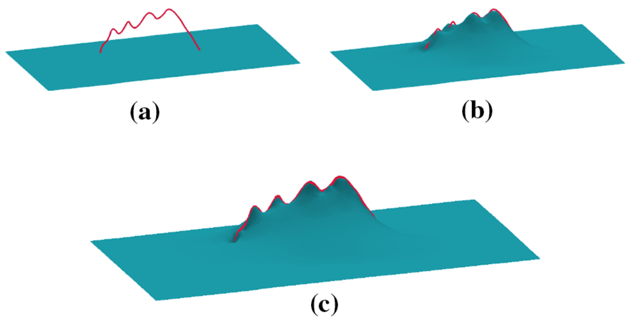

Fig. 8 The terrain deformation technique at multiple resolutions. **a** The input terrain T_0 and control curve P. **b** The deformed terrain T_1 after one level of the subdivision. **c** The deformed terrain T_3 after three levels of the subdivision

To provide a good fit for detailed curves, our technique must capture the curve's details at several resolutions. To address this issue, we not only move the initial vertices, but also solve an optimization problem for the vertices replacements at each resolution to capture the curve's fine details. Therefore, as the terrain's resolution increases, the terrain approximates details which could not be captured at lower resolutions. Accordingly, for each resolution t, given the terrain T_t, we place the vertices V^t such that it minimizes the distance between the control curve and the subdivided terrain:

$$\min_{\Delta_t} d(S(T_t + \Delta_t), P) \tag{1}$$

where Δ_t is a perturbation vector for V^t, $S(T)$ denotes subdivision of T, and d is the distance between P and the subdivided terrain. The distance between the subdivided terrain and the points $p_j \in P$ of the control curve is computed using the distance between p_j and its projection p_j^{t+1} onto the subdivided terrain. The projection p_j^{t+1} falls inside a triangle with vertices v_a^{t+1}, v_b^{t+1} and v_c^{t+1} and can be written as:

$$p_j^{t+1} = \alpha v_a^{t+1} + \beta v_b^{t+1} + \gamma v_c^{t+1} \tag{2}$$

where α, β and γ are the barycentric coordinates. Therefore, to minimize Eq. 1, we minimize $\sum \| p_j^{t+1} - p_j \|$ for $p_j \in P$ where p^{t+1} is defined in Eq. 2. This produces the following positional constraints:

$$p_j^{t+1} = p_j, \text{ for } j \in \{1, 2, \ldots, m\}. \tag{3}$$

which, we can rewrite as a function of V^t using Eq. 2. As our subdivision mask is a linear operator, the position of every vertex v_i^{t+1} can be written as $v_i^{t+1} = \alpha_1 v_1^t + \alpha_2 v_2^t + \cdots + \alpha_n v_n^t$, where n is the number of vertices at resolution t, and the coefficient α_j is defined by S_i (i.e. the ith row of the subdivision matrix S). Therefore, a positional constraint can be rewritten to depend on the vertices V^t:

$$\begin{aligned} p_j^{t+1} &= \alpha v_a^{t+1} + \beta v_b^{t+1} + \gamma v_c^{t+1} \\ &= \alpha S_a V^t + \beta S_b V^t + \gamma S_c V^t \\ &= [\alpha S_a + \beta S_b + \gamma S_c] \left[v_1^t \; v_2^t \; v_3^t \; \ldots \; v_n^t \right]^T , \end{aligned} \tag{4}$$

yielding a banded linear system of equations relating P and V^t.

The positional constraints form an overdetermined system, and the minimizer of this system is computed by solving a least-squares problem (i.e using the pseudo-inverse). Figure 8 shows an example of employing this method to deform a flat terrain. In the next sections, the preceding method is extended to a set of specified features.

3.2.2 Terrain deformation based on a set of features

Here we extend the previous method to approximate not only the control curve, but also a feature slopes and field of influence. Our goal is to deform the terrain to best fit the control curves, associated slopes and fields of influence. Figure 4a depicts an input feature in which the control curve (red curve) specifies the feature and elevation along it, and the cross section curve (orange curve) specifies the feature's slope and field of influence. To approximate the slope and field of influence along the control curve, we define extra positional constraints based on the cross section curve. To impose these constraints along the control curve, the cross section curve is translated and scaled at regular intervals along the control curve, and oriented using the rotation minimizing frame [57] as shown in Fig. 4a. Thus, given a generated cross section curve defined by the polyline constructed from a set of 3D points denoted as $C = \{c_1, c_2, \ldots, c_l\}$, extra positional constraints can be defined as:

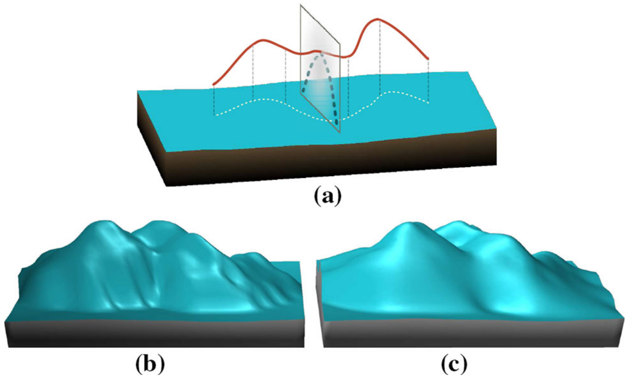

Fig. 9 An example of the terrain deformation with and without energy minimization constraints. **a** The input feature and terrain. **b** The deformed terrain only based on the positional constraints. **c** The deformed terrain using the positional and energy minimization constraints

$$c_j^{t+1} = c_j, \text{ for } j \in \{1, 2, \ldots, l\}, \tag{5}$$

where c_j^{t+1} is the projection of c_j onto the subdivided terrain (Eq. 2) (Fig. 4b).

The control and cross section curves do not have the same importance in the resulting least-squares problem, as the control curve is more accurately specified in orthophotos. To address this issue, we impose the positional constraints of the cross section curves after determining locations of vertices based on the control curve, as suggested by Hnaidi et al. [20]. This gives rise to two least-squares problems which determine the positions of V^t. In the first problem, we compute the positions of vertices V^t that are affected by the constraints defined in Eq. 3, and in the second problem, by fixing the positions of the vertices in the previous step, we compute the positions of vertices based on Eq. 5 (Fig. 4b). Additionally, dividing the problem into two subproblems reduces the size of the least-squares problem and increases computational efficiency.

Editing of the terrain purely based on positional constraints can result in high curvature areas due to non-regularized least-squares solutions [47]. Furthermore, moving a subset of the terrain's vertices without considering the adjacent vertices can result in high curvature areas at the boundary of the deformed region (Fig. 9). To address these issues, we introduce a constraint to minimize the curvature of the deformed region. To approximate surface curvature at a vertex, we use the discrete Laplace–Beltrami operator [12]. Thus, a energy minimization constraint for each vertex v_j^{t+1} is defined as:

$$L_j^{t+1} = v_j^{t+1} - \frac{1}{d_j} \sum_{v_i^{t+1} \in N(v_j^{t+1})} v_i^{t+1} = 0,$$

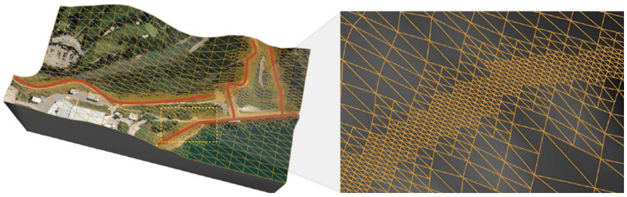

Fig. 10 An example showing the adaptively subdivided terrain based on the features. The *left image* illustrates the provided features on the terrain, and the *right* one depicts the adaptively subdivided terrain around a feature

where d_j and $N(v_j^{t+1})$ are the degree and adjacent vertices of v_j^{t+1}. To eliminate high energy behaviors, we impose the energy minimization constraint on all the vertices that are affected by the control and cross section curves or falls within a specified distance from the control curve (Fig. 4b). The above constraint is considered along with the positional constraints for the vertices relocated by both least-squares problem.

As mentioned earlier, to approximate features and associated characteristics at each resolution, our technique must be iteratively applied to the result of optimizing the previous resolution to reach adequate precision. By applying the method repeatedly, as the number of vertices at the base terrain and the size of least-square system increase, we obtain higher accuracy around features. Finally, this approach can be extended to a set of features by computing the positional and energy minimization constraints based on all control and cross section curves simultaneously.

Since features may only affect a small region on DEM, we avoid increasing the resolution for the entire region of interest. To increase details around features, we adaptively subdivide the terrain (Fig. 10) by employing incremental adaptive Loop subdivision [44]. As discussed by Pakdel and Samavati, adaptive subdivision techniques have some shortcomings which must be handled delicately [44]; otherwise, skinny triangles, cracks or abrupt change of resolution may appear in the resulting DEM. Continuous change of details makes rendering DEMs at different resolution possible with a simple and efficient technique such as zero area triangles [37]. In our system, to preserve the hierarchy of DEM and support continuous change of details, terrain is adaptively subdivided such that adjacent triangles must be within one level of each other in the terrain hierarchy.

4 Bodies of water tool

Representing bodies of water is important for many DE environmental applications which require monitoring, visualizing and simulating water bodies [39]. However, acquisition techniques of Digital Elevation Model are mostly unable to

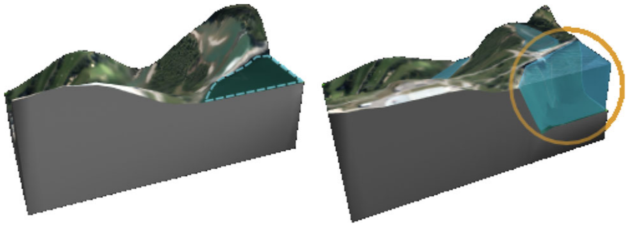

Fig. 11 An example showing the application of the body of water tool. The *left image* shows the input terrain and the boundary of the body of water, and the *right image* depicts the body of water created by employing our tools

capture the underlying structure of rivers, lakes and sea beds. In our system, bodies of water can be created interactively using a simple sketch-based tool. To use this tool, first terrain has to be edited to create a basin (see Fig. 11). Second, the user draws a closed stroke onto the terrain corresponding to the water body boundary. Our system then automatically generates the body of water based on the elevations of vertices inside and around the region enclosed by the stroke.

5 Vegetation tool

Orthophotos provide some information regarding the plant ecosystems present in a given terrain, and augmenting DE representations with plant models substantially increases their accuracy and realism in 3D scenes. However, these photos are typically insufficient for the detailed 3D reconstruction of individual trees and shrubs. On the other hand, they provide vast amounts of information regarding the placement, distribution and color of plants. Accordingly, our system provides a sketch-based tool guided by an orthophoto to create vegetation on terrain. As initial data, 3D models of trees and plants are retrieved from a database (in a DE this would be based on the region of interest or commonly available vegetation species diversity). As shown in Fig. 12, the region containing plants and vegetation is specified interactively by sketching a closed stroke onto the terrain based on the orthophoto (cyan stroke). Similar to [28], plants are distributed onto the region based on the average distance between the input plants in the region. The average distance can be provided either automatically [60] or interactively by the user. The created plants are colored based on the orthophoto to create a plant ecosystem with a similar visual character to that present in the selected region. Therefore, the top view of the terrain with vegetation remains consistent with the orthophoto (see Fig. 12).

To distribute plants onto the specified area, we start by projecting strokes onto the terrain, and triangulating the 3D polygon with respect to DEM data using *Delaunay triangulation* [9]. Afterwards, plants are randomly distributed onto the region with respect to the areas of triangles (i.e. larger triangles get more plants than smaller ones). The number of plants for each input model is determined based on the average distance specified for the region. Finally, leaves are colored based on the orthophoto. Our tool considers small neighborhood around each position to determine leaf color. Furthermore, plants of the same type are randomly scaled and rotated to create more variation.

6 Terrain texturing using orthophotos

Orthophotos capture many aspects of the visual appearance of features. Consequently, using them to texture the terrain enhances its visual appearance greatly. Since many features are represented in orthophotos, it is particularity beneficial to have a set of smart image editing tools to modify them. Our system includes a clone brush for editing these images. The clone brush can be used for removing unwanted regions. For instance, a 3D object such as a bridge is not part of the geometry of the terrain, so its footprint and shadow must be removed and replaced by terrain material to be used as texture (see Fig. 13). Clone brush is also useful for repairing the texture of objects that are obscured by occlusion.

This tool can also be used for cloning features, such as vegetation or bodies of water, to create a new image which can be used later as a guide for modeling new landscapes (see Fig. 14). Figure 14a, b illustrates the original and modified image, respectively. As demonstrated in Fig. 14b, using our clone brush, landscape elements have been modified to create a new

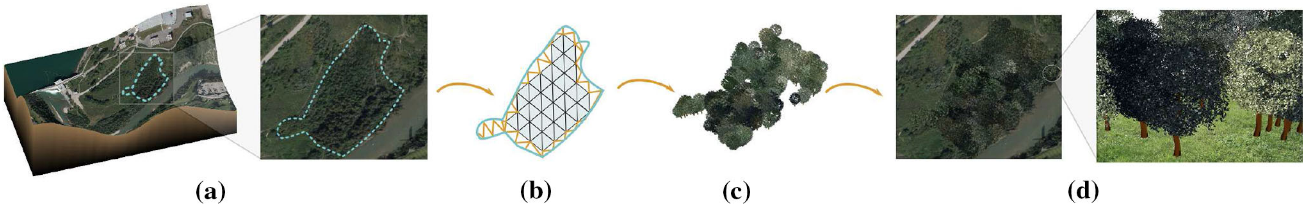

Fig. 12 An example demonstrating the vegetation tool. **a** The input stroke around the region covered with vegetation. **b** Triangulating the projected stroke with respect to DEM data. **c** Generating plants based on the orthophoto and selected region. **d** Vegetation shown on the terrain. The *top view* of the terrain with vegetation remains consistent with the orthophoto. The *right image* shows a closer view of the created scene

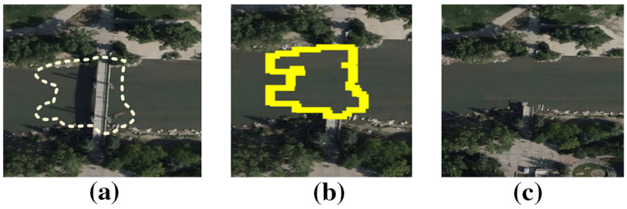

Fig. 13 An example showing the application of Clone brushing tool for removing unwanted regions. **a** Sketching the unwanted region in the original image. **b** The modified image after cloning. All the pixels up to a specific distance *d* from the boundary are colored in *yellow*. **c** The final result after synthesizing the boundary pixels of the unwanted region

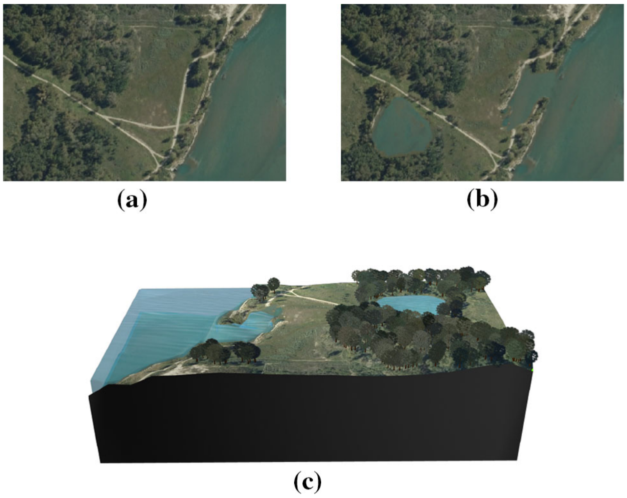

Fig. 14 A novel landscape generated on the basis of a preexisting terrain. **a** The input orthophoto. **b** The modified orthophoto using our clone brush. **c** The result after generating the new landscape from the modified photo

scene. Finally, Fig.14c presents the result after creating new landscape elements interactively using our tools. This feature is particularly beneficial for landscape planning applications.

There are two challenges regarding cloning a portion of an image to another region. Copying information from one part of an image to another can result in distortion at the boundary of the selected region. To minimize distortion around the boundary, all the pixels up to a specific distance *d* from the boundary are synthesized based on the inside and outside regions (see Fig. 13b). To have a fast real-time tool, we use the texture synthesizer proposed by Simakov [51], and for finding the best patch we apply PatchMatch [2].

Furthermore, our clone brush considers the underlying terrain geometry as opposed to the traditional image processing tool. For instance, sloped terrain causes texture foreshortening. Therefore, to avoid unrealistic distortion during cloning, our tool adaptively resizes the destination region based on terrain slopes at the source and destination.

7 3D structure reconstruction tool

The 3D reconstruction of urban areas and man-made structures, in particular, has received an increasing amount of interest in recent years [40], but fully automatic algorithms have not been completely successful in reconstructing 3D urban models [40]. Therefore, interactive methods, which can support the creation of 3D man-made structures, have gained importance within the context of DE. In this section, we focus on interactive 3D modeling of man-made structures, as a crucial aspect of urban areas, for Digital Earth.

To enhance the availability of existing data while maintaining consistency with other aspects of our system, we propose extracting 3D man-made structures from available photos. We suggest an easy and novel interactive technique to extract textured 3D structures via a few user interactions and integrate them into our 3D maquette using simple sketch-based interactions.

Different types of photos can be used to extract 3D structures. Orthophotos as an input image provide the roofs of these structures, which can potentially be used for 3D reconstruction. A possible approach to create such a 3D model is to specify a building footprint interactively and extrude it based on the building's height. Figure 15 depicts an example of 3D buildings resulting from employing this approach. The roofs of the buildings are textured using the footprints specified in the orthophotos, and the geometry of their bases is determined based on the underlying terrain. This approach can be also extended to more complex architectures, as discussed in the work of Kelly and Wonka [24]. Due to the lack of detail in the created models, Kelly and Wonka [24] suggest adding details using procedural modeling. An advantage of extruding 3D buildings from orthophotos is that the created models match the scale and orientation of their footprints after the extrusion.

Although this technique is useful for generating 3D building models, it does not support some types of 3D structures such as bridges and overpasses. In fact, these structures are not completely attached to the ground, and are typically built on sloped terrain. Consequently, more accurate reconstruction of these structures can eliminate unrealistic distortions such as those depicted in Fig. 16a. To extract these structures

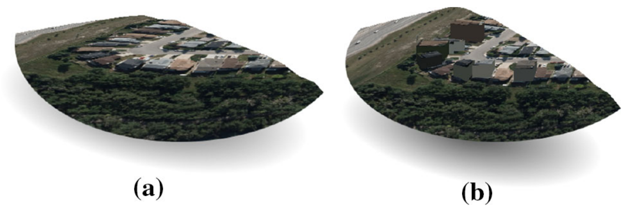

Fig. 15 An example of the creation of 3D buildings by extruding their footprints. **a** The textured terrain. **b** The created 3D buildings using their footprints

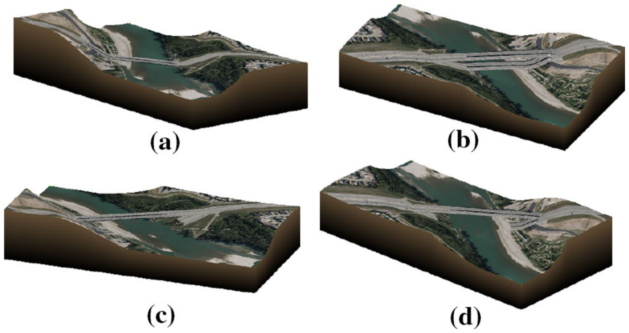

Fig. 16 An example of extracting a bridge from an orthophoto. **a** The input terrain. **b** The extracted bridge on the terrain. **c**, **d** The result after employing the clone brush

from orthophotos, we developed a simple technique in which, by specifying two edges of a bridge or overpass which are not attached to the terrain, we extract the geometry of these models based on the underlying terrain at the intersection points. Figure 16 illustrates an example of extracting a bridge using our technique. By employing our technique, we extract the bridge, as depicted in Fig. 16b. To remove the bridge footprint and its shadow from the orthophoto, we applied our clone brush. Figure 16c, d shows the results from two different viewpoints.

Although orthophotos have been considered for 3D reconstruction [29], they have a number of limitations as input data. The geometry of many structures cannot be fully determined based on their tops (e.g. bridges and dams). Additionally, since orthophotos do not provide height information, these values must be manually adjusted for reconstructions. Furthermore, orthophotos provide little or no information about the sides of a structure. Therefore, other techniques must be utilized to texture the object [31].

On the other hand, a single photo such as that shown in Fig. 3 is a good complementary piece of information, providing both the geometry and texture for man-made structures. These reference photos are not less detailed than ground view images; however, they cover a wide area and contain a variety of structures. These types of photos are particularly useful for extracting multiple 3D structures with a few interactions. Since the reconstruction of 3D models from a single photo is an ill-posed problem, some basic input from the user has proven to be essential [42]. Therefore, we propose an interactive technique for the extraction and integration of a set of 3D structures from a photo. Correspondingly, our proposed system supports loading an available image as a reference for this process (Fig. 3).

Assuming some simple properties about these structures makes 3D reconstruction from a single image possible when augmented by guidance from the user through some interactions. As observed by Kosecka and Zhang [26], man-made structures are typically composed of orthogonal rectangular shapes which can be utilized to reconstruct them from a single view. In this regard, one of the main challenges is the issue of perspective distortion in input photos. An interactive object extraction technique based on the orthogonality of edges was introduced in the work of Chen et al. [8]. They propose a fast technique for extracting 3D objects, in the form of general cylinders and cuboids, based on three orthogonal axes. Due to the simplicity and efficiency, we extend their technique to extract 3D man-made structures from a single view. Our technique is designed to extract the 3D geometry and facade texture of structures in the form of *right prisms*, which have at least one right angle corner visible in the photo. Figure 17 depicts the paradigm used to define 3D structures. In Fig. 17, the right angle corner is specified by a blue circle. By finding three orthogonal lines with the help of the user, our system can compute the 3D geometry of a structure by removing the perspective effect. Additionally, our method enables the integration of 3D models into a Digital Earth framework with a few interactions.

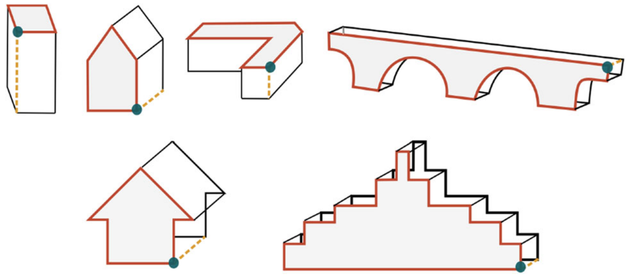

Fig. 17 Examples of supported 3D structures

7.1 Sketch-based interaction

Figure 18 shows the extraction of several objects from a reference image. As demonstrated in Fig. 18a, each building is extracted by sketching three strokes. As depicted in Fig. 18b, c, our technique preserves the relative heights of structures. We also use the ground direction to determine a consistent vertical alignment.

As illustrated in Fig. 19a, some structures are not simple rectangular prisms. To support more general prisms, a visible planar side of the structure (*profile*) is interactively provided in the photo (red polygon) (Fig. 19b). The profile must have at least one right angle. Finally, an edge of the structure orthogonal to the profile is specified by sketching a stroke (shown in orange). As a result, the textured 3D structure is automatically extracted from the image (Fig. 19b, c). In the next section, we describe our extraction method based on the orthogonal edges provided by the user.

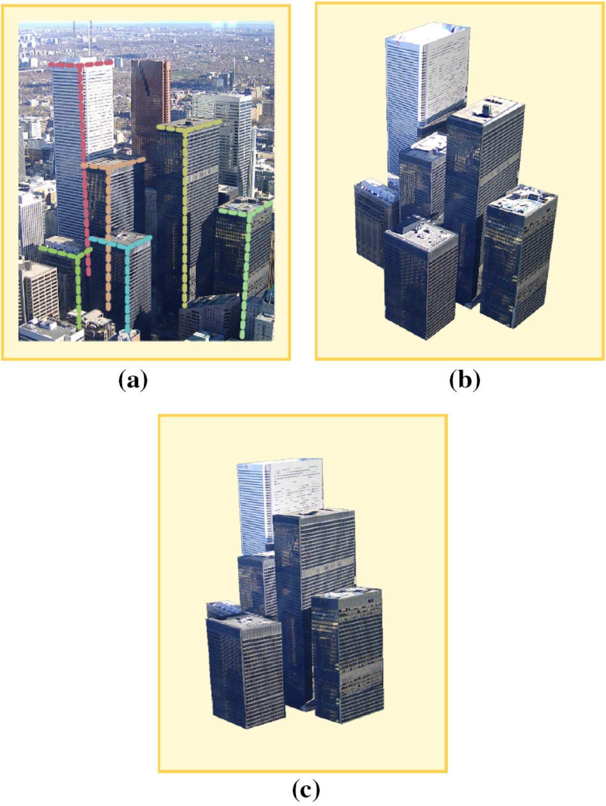

Fig. 18 Extracting symmetric objects from a single image. **a** Specifying three orthogonal strokes. **b** Extracted 3D objects. **c** Extracted 3D objects from a different viewpoint

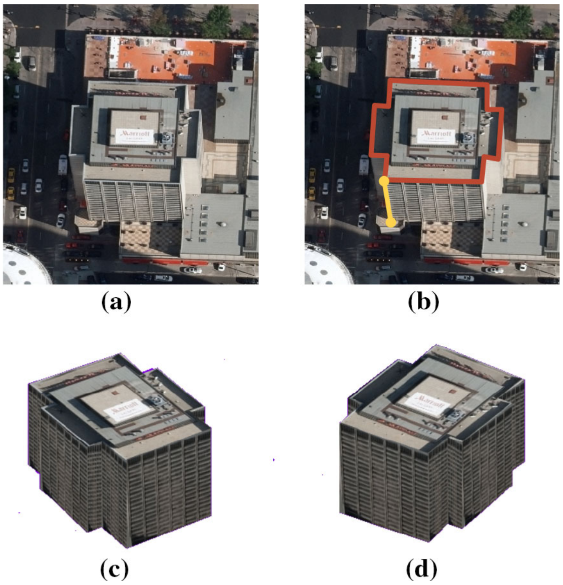

Fig. 19 An example of extracting a 3D structure by specifying its profile. **a** The input image. **b** The profile (in *red*) and the line orthogonal to it (in *yellow*). **c**, **d** The result after extracting the model

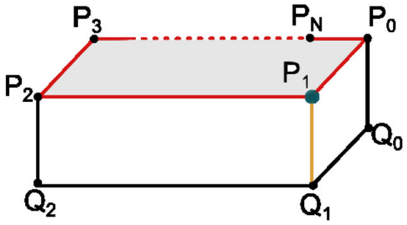

Fig. 20 Computing the positions of vertices

7.2 Extraction method

To extract a 3D structure from an image, we must determine the world coordinates of its points based on their 2D image coordinates. Figure 20 depicts a right prism where the vertices of its two profiles are denoted by $\{P_0, P_1, \ldots, P_N\}$ and $\{Q_0, Q_1, \ldots, Q_N\}$, and the lines connecting corresponding vertices $(P_i Q_i)$ are orthogonal to both profiles. The projections of the two profiles on the image plane are denoted by $\{P'_0, P'_1, \ldots, P'_N\}$ and $\{Q'_0, Q'_1, \ldots, Q'_N\}$ in 2D. In addition, P_1 in Fig. 20 denotes a 3D point where three orthogonal lines $\{P_2P_1, P_0P_1, Q_1P_1\}$ intersect. By specifying the projection of these lines on the image plane $\{P'_2P'_1, P'_0P'_1, Q'_1P'_1\}$ interactively, we can find the 3D geometry of the object by computing the world coordinates of all the points based on the orthogonality constraints:

$$\begin{cases} (P_2 - P_1).(P_0 - P_1) = 0 \\ (P_0 - P_1).(Q_1 - P_1) = 0 \\ (Q_1 - P_1).(P_2 - P_1) = 0 \end{cases} \tag{6}$$

To compute the world coordinates of these points, we must find the relation between its image coordinates and its world coordinates. The 2D image coordinate $\{x', y'\}$ of a point is determined via a perspective projection of its world coordinate $\{x, y, z\}$. In this regard, understanding the effects of perspective projection is crucial for finding the relation between $\{x, y, z\}$ and $\{x', y'\}$. As a result of perspective projection, distant structures, in the image, appear smaller than structures closer to the camera. Therefore, parallel lines in the world coordinate system intersect in the image.

To find the relation between $\{x, y, z\}$ and $\{x', y'\}$, we use a simplified camera model (zero skew and no radial distortion) [59]. The perspective projection of this camera is defined as:

$$M = \begin{bmatrix} f & 0 & u & 0 \\ 0 & f & v & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \tag{7}$$

where f is the focal length, and (u, v) is the principle point in the image coordinate system. By applying M to the world coordinates, $\{x', y'\}$ can be represented in terms of $\{x, y, z\}$ as:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = M \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}.$$

For example, by considering the simplest form of perspective projection ($f = 1$ and $(u, v) = (0, 0)$), (x', y') is equal to $(\frac{x}{z}, \frac{y}{z})$. Accordingly, $\{x, y\}$ can be rewritten in terms of $\{x', y'\}$:

$$(x, y) = \left(\frac{(x' - u)z}{f}, \frac{(y' - v)z}{f}\right). \tag{8}$$

Therefore, given Eq. 8, the orthogonality constraints (Eq. 6) can be rewritten in terms of image coordinates:

$$\left(\frac{(x'_m - u)z}{f} - \frac{(x'_r - u)z}{f}\right)\left(\frac{(x'_n - u)z}{f} - \frac{(x'_r - u)z}{f}\right)$$
$$+\left(\frac{(y'_m - v)z}{f} - \frac{(y'_r - v)z}{f}\right)\left(\frac{(y'_n - v)z}{f} - \frac{(y'_r - v)z}{f}\right)$$
$$+(z_m - z_r)(z_n - z_r) = 0,$$

where

$$\begin{bmatrix} x_m \\ y_m \\ z_m \end{bmatrix}, \begin{bmatrix} x_n \\ y_n \\ z_n \end{bmatrix} \subset \{P_0, P_2, Q_1\} \quad \text{and} \quad \begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix} = P_1. \tag{9}$$

By considering f and (u, v) as constants [8], Eq. 9 has four unknowns, which are z values of $\{P_0, P_1, P_2, Q_1\}$, and three independent equations. As suggested in the work of Chen et al. [8], if the z value of P_1 is set to a constant, then the z values of the other points can be computed based on Eq. 9, and correspondingly the z values of $\{P_0, P_1, P_2, Q_1\}$ may be computed based on Eq. 8. Therefore, given the z values of P_0, P_1, P_2, we can compute the z values of other points $\{P_2, P_3, \ldots, P_N\}$.

Finally, the position of Q_i is determined by translating P_i in the direction of $Q_0 - P_0$. Once the world coordinates of the points are determined, we can deduce the texture of the structure based on the image coordinates. In the next section, we describe our method for texturing the extracted 3D model based on the image coordinates.

7.3 Texture

Input data, such as the photo depicted in Fig. 18, provide partial texture information for the extracted objects. However, occlusions in the image can lead to missing texture information. To texture the object, first we have to find the visible facets in the photo. For mapping texture to the facets, we use the corresponding image coordinates of the points. Once we compute the texture coordinates of the points, we can find non-occluded facets by applying a visibility test. For each facet, we cast a ray toward the center of the facet in a direction orthogonal to the image plane. If the ray does not hit the facet first, it is marked as a hidden facet.

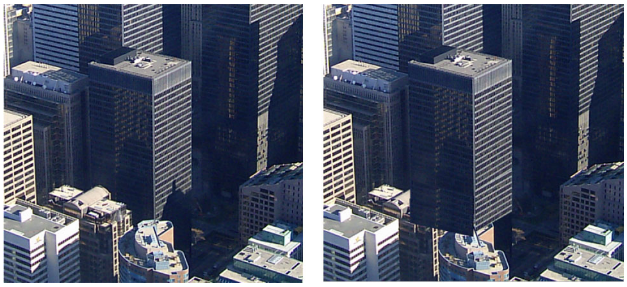

Fig. 21 To resolve occlusions which cannot be resolved based on symmetry, the user can use the clone brush to duplicate portion of the image for repairing occluded areas. The *left* and *right panels* depict the original and modified textures

To texture the hidden facets, we exploit the assumed symmetry of the objects by copying the texture of visible facet to hidden ones. This can be accomplished by finding the most similar visible facets parallel to the hidden one. Furthermore, to resolve occlusions which cannot be resolved based on symmetry, the user can use the clone brush (Sect. 6) to duplicate portions of the image to repair occluded areas (Fig. 21). Once a textured 3D model is extracted from a photo, it has to be integrated into a DE framework. In the next section, we discuss a sketch-based technique designed to integrate 3D structures.

7.4 Integrating objects into a Digital Earth framework

As illustrated in Fig. 22, an extracted 3D object has to be integrated into a specific location on a 3D maquette (for export to DE) by rotating, translating and scaling the object to match its footprint in the orthophoto. Applying these transformations in 3D using traditional interfaces, in which translation, rotation and scaling are three different operations, is often challenging and time-consuming. To simplify this process, we adapt the technique proposed by Severn et al. [50]. We introduce a sketch-based method for integrating the object into the geographic coordinate system by sketching a single stroke (transformation stroke) around its footprint onto the terrain (see Fig. 23). This stroke is used to determine the instancing transformation (i.e rotation, scaling, translation).

As depicted in Fig. 22, the 3D object in the image coordinate system $\{u, v, w\}$ has to be transformed into the geographic coordinate system $\{u', v', w'\}$. The translation, rotation and scaling are determined based on the shape of the

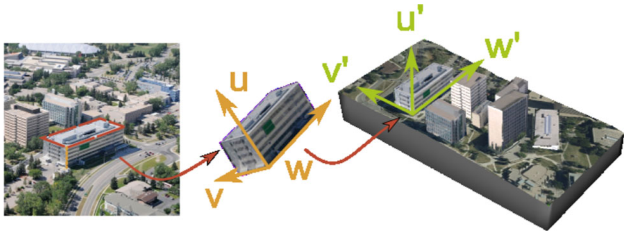

Fig. 22 Integrating 3D structures into DE. **a** The input image. **b** The strokes required for extracting the 3D model. **c**, **d** The extracted model and the transformation stroke on the terrain. **e** The integrated model on the 3D maquette

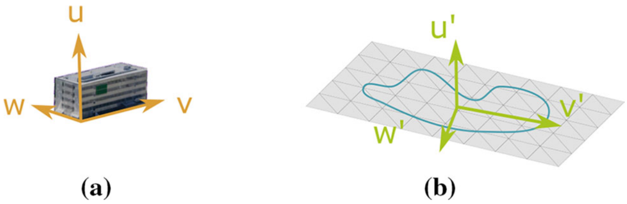

Fig. 23 Once u and u' are aligned, we can use the transformation stroke to align the other two axes. This alignment can be done by extracting two main directions (v' (major axis) and w' (minor axis)) from the transformation stroke using Principle Component Analysis

stroke (Fig. 22d (blue stroke)). The translation can be done easily by moving the center of the 3D object to the center of the stroke, and placing it on the ground by projecting its bottom vertices onto the terrain. The object also has to be rotated to align vector u to u' (ground direction).

Once u and u' are aligned, we can use the transformation stroke to align the other two axes (Fig. 23). This alignment is achieved by extracting two main directions (v' (major axis) and w' (minor axis)) from the transformation stroke using Principle Component Analysis [13] (see Severn et al. [50] for details). The same approach can also be used to determine two main directions (v and w) of vertices at the bottom of the 3D object. Once v and v' are computed, the object is rotated to align v to v'. Finally, as mentioned in the work of Severn et al. [50], the magnitude of the major and minor axes can be used to scale the object, such that it fits its footprint in the orthophoto.

For cases where the transformation cannot be uniquely identified, such as symmetric structures, the object can be transformed interactively using conventional methods. Since u is already aligned with u', a 2D transformation suffices to place the object with the same orientation as the footprint.

8 Results

To illustrate the methods presented in previous sections, we implemented a sketch-based system which supports a variety of landscapes. As input data, we used DEMs available from the *US Geological Survey*, and orthophotos from the *City of Calgary* datasets. We present an example of the creation and editing of landscape by considering the Glenmore reservoir located in the southwest quadrant of Calgary, Alberta (Fig. 24). The input data and generated contents are individually depicted in Fig. 3.

Figure 24a, b illustrates the input orthophoto and terrain, respectively. Features such as the river, reservoir and roads in the orthophoto are not accurately represented in the input DEM. To correct the input DEM, four features are specified in the orthophoto. To create bodies of water, three closed strokes are sketched onto the terrain. Vegetation is created based on four strokes around the areas covered by plants. The final result is illustrated in Fig. 24c, d. To export this information for a height-map based DE framework, the terrain hierarchy is represented in height map format for each resolution.

Figure 25 depicts another example of the creation and editing of landscape by considering Elliston lake located in the southeast quadrant of Calgary, Alberta (Fig. 25). Figure 25a, b illustrates the input orthophoto and terrain, respectively. Some features including the lake and roads are not accurately represented in the underlying DEM data. Therefore, to correct the geometry of the terrain, two features (the lake and one of the roads) are specified in the orthophoto, and the body of water is created by sketching a closed stroke around the boundary of the lake. Vegetation is specified and created based on six strokes around the areas covered by plants.

Additionally, our system supports the integration of new designs and ideas into a DE representation. As illustrated in Fig. 14, by modifying an orthophoto using our tools, a new landscape can be modeled and explored in 3D. Our system creates a platform for the setup, analysis and visualization of new concepts within the context of DE.

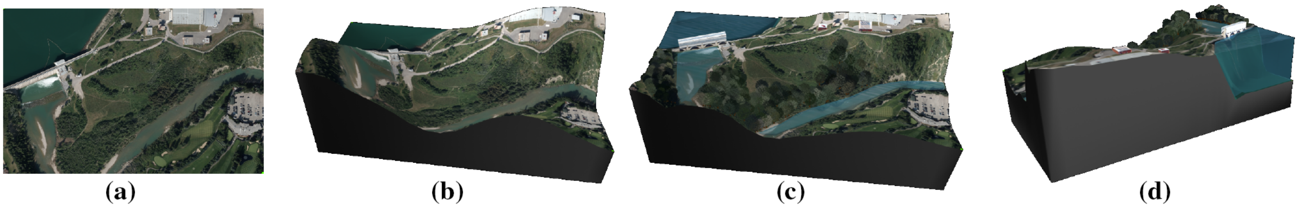

Fig. 24 Glenmore reservoir. **a** Input orthophoto. **b** Original terrain. **c** Result after using the proposed system. **d** Result from a different viewpoint

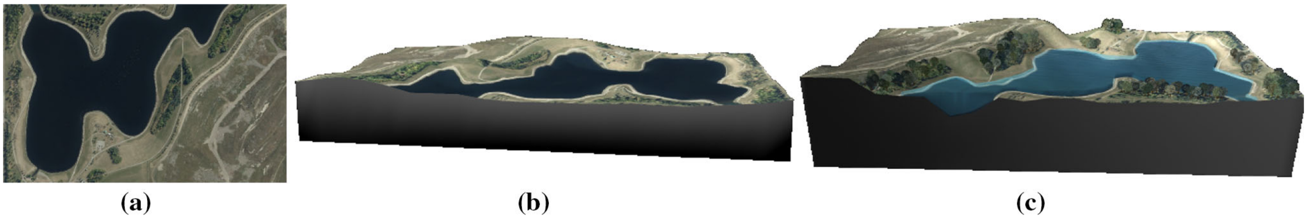

Fig. 25 Elliston Regional Park. **a** Input orthophoto. **b** Original terrain. **c** Result after using the proposed system

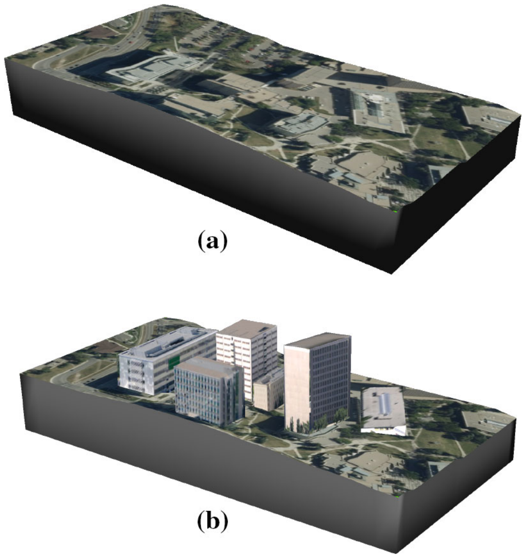

Fig. 26 University of Calgary campus. **a** Original terrain. **b** Result after the creation and integration of several 3D structures

Our system is also useful in the creation of 3D content for urban areas. Figure 26 depicts University of Calgary campus. It shows the original terrain without any 3D structures (Fig. 26a) in comparison with new 3D content created and integrated into the geographic coordinate system (Fig. 26b). All these 3D buildings are extracted from the single image depicted in Fig. 22. The extraction and integration of each 3D structure require three lines and a transformation stroke, and the entire process takes only five minutes.

9 Conclusion

In this paper, we introduce a sketch-based system for creating 3D contents from a single photo and enhancing the quality of existing data in a DE framework. Our system is capable of creating a wide range of landscapes from limited input data, such as a low quality DEM and an orthophoto.

There are several directions in which this work can be extended. For generating plants ecosystem based on an orthophoto, the density of plants could potentially be obtained via frequency analysis of an orthophoto. Currently, the user provides the average distance between plants in the photo. To make our system simpler and more interactive, it could support snapping and flood fill operations for specifying features such as rivers and edges of structures [42]. Additionally, the approach for extracting 3D structures from a single image can be improved by considering multi-part structures [23]. This would improve on the current implementation, where the user extracts different parts of a structure individually and integrate them into a maquette.

Acknowledgments We would like to deeply thank Kaveh Ketabchi for his numerous contributions to this paper. This work was supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada.

References

1. Agarwal, S., Furukawa, Y., Snavely, N., Simon, I., Curless, B., Seitz, S.M., Szeliski, R.: Building rome in a day. Commun. ACM **54**(10), 105–112 (2011)
2. Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.: Patchmatch: a randomized correspondence algorithm for structural image editing. ACM Trans. Graph. TOG **28**(3), 24 (2009)
3. Bernhardt, A., Maximo, A., Velho, L., Hnaidi, H., Cani, M.P.: Realtime terrain modeling using cpu-gpu coupled computation. In: 24th SIBGRAPI Conference, pp. 64–71. IEEE (2011)
4. Bolstad, P.: Gis fundamentals. A first text on Geographic Information Systems (2005)
5. Botsch, M., Sorkine, O.: On linear variational surface deformation methods. IEEE Trans. Visual. Comp. Graph. **14**(1), 213–230 (2008)
6. Brosz, J., Samavati, F., Sousa, M.: Terrain synthesis by-example. Adv. Comp. Graph. Comp. Vision **4**, 58–77 (2007)
7. de Carpentier, G.J., Bidarra, R.: Interactive gpu-based procedural heightfield brushes. In: Proceedings of the 4th International Conference on Foundations of Digital Games, pp. 55–62. ACM (2009)
8. Chen, T., Zhu, Z., Shamir, A., Hu, S.M., Cohen-Or, D.: 3-sweep: extracting editable objects from a single photo. ACM Trans. Graph. (TOG) **32**(6), 195 (2013)
9. De Berg, M., Van Kreveld, M., Overmars, M., Schwarzkopf, O.C.: Computational Geometry. Springer, Berlin (2000)
10. Debevec, P.E., Taylor, C.J., Malik, J.: Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach. SIGGRAPH '96, pp. 11–20. ACM (1996)
11. Deussen, O., Hanrahan, P., Lintermann, B., Měch, R., Pharr, M., Prusinkiewicz, P.: Realistic modeling and rendering of plant ecosystems. SIGGRAPH '98, pp. 275–286. ACM (1998)
12. Do Carmo, M.P.: Differential Geometry of Curves and Surfaces, vol. 2. Prentice-hall, Englewood Cliffs (1976)
13. Duda, R.O., Hart, P.E., et al.: Pattern Classification and Scene Analysis, vol. 3. Wiley, New York (1973)

14. Gain, J., Marais, P., Straßer, W.: Terrain sketching. In: Proceedings of the 2009 symposium on Interactive 3D graphics and games, pp. 31–38. ACM (2009)
15. Gao, K., Rockwood, A.: Multi-sided attribute based modeling. In: Proceedings of the 11th IMA International Conference on Mathematics of Surfaces, IMA'05, pp. 219–232. Springer-Verlag, Berlin, Heidelberg (2005). doi:10.1007/11537908_13
16. Goodchild, M.F.: Discrete global grids for digital earth. In: Proceedings of 1st International Conference on Discrete Global Grids (2000)
17. Gore, A.: The digital earth: understanding our planet in the 21st century. Aust. Surv. **43**(2), 89–91 (1998)
18. Habbecke, M., Kobbelt, L.: Linear analysis of nonlinear constraints for interactive geometric modeling. In: Computer Graphics Forum. **31**, pp. 641–650. Wiley Online Library (2012)
19. Hammes, J.: Modeling of ecosystems as a data source for real-time terrain rendering. In: Digital Earth Moving, pp. 98–111. Springer (2001)
20. Hnaidi, H., Guérin, E., Akkouche, S., Peytavie, A., Galin, E.: Feature based terrain generation using diffusion equation. In: Computer Graphics Forum. **29**, pp. 2179–2186. Wiley Online Library (2010)
21. Irschara, A., Zach, C., Bischof, H.: Towards wiki-based dense city modeling. In: Computer Vision, 2007, ICCV 2007, IEEE 11th International Conference on, pp. 1–8 (2007)
22. Jenny, H., Jenny, B., Cartwright, W.E., Hurni, L.: Interactive local terrain deformation inspired by hand-painted panoramas. Cartograph. J. **48**(1), 11–20 (2011)
23. Jiang, N., Tan, P., Cheong, L.F.: Symmetric architecture modeling with a single image. ACM Trans. Graph. (TOG), **28**, 113 (2009)
24. Kelly, T., Wonka, P.: Interactive architectural modeling with procedural extrusions. ACM Trans. Graph. (TOG), **30**(2), 14 (2011)
25. Ketabchi, K., Runions, A., Samavati, F.: 3d maquetter: sketch-based 3d content modeling for digital earth. In: International Conference on Cyberworlds (CW) 2015, pp. 1–9. IEEE Computer Society (2015)
26. Košecká, J., Zhang, W.: Extraction, matching, and pose recovery based on dominant rectangular structures. Comp. Vision Image Underst. **100**(3), 274–293 (2005)
27. Lafarge, F., Mallet, C.: Building large urban environments from unstructured point data. In: 2011 IEEE International Conference on Computer Vision (ICCV), pp. 1068–1075. IEEE (2011)
28. Lane, B., Prusinkiewicz, P., et al.: Generating spatial distributions for multilevel models of plant communities. In: Graphics Interface, pp. 69–80. Citeseer (2002)
29. Lee, S.C., Huertas, A., Nevatia, R.: Modeling 3-d complex buildings with user assistance. In: Fifth IEEE Workshop on Applications of Computer Vision, 2000. pp. 170–177. IEEE (2000)
30. Lee, S.C., Jung, S.K., Nevatia, R.: Automatic pose estimation of complex 3d building models. In: Proceedings. Sixth IEEE Workshop on Applications of Computer Vision, 2002, (WACV 2002), pp. 148–152. IEEE (2002)
31. Lee, S.C., Jung, S.K., Nevatia, R.: Integrating ground and aerial views for urban site modeling. In: Proceedings of 16th International Conference on Pattern Recognition, 2002, vol. **4**, pp. 107–112, IEEE (2002)
32. Lee, S.C., Nevatia, R.: Interactive 3d building modeling using a hierarchical representation. In: Higher-Level Knowledge in 3D Modeling and Motion Analysis., pp. 58–65. IEEE (2003)
33. Lee Staff, J.S., Angawi, S., Angawi, A., Rockwood, A.: A new terrain modeling technique for digital reconstruction–a case study. In: Proceedings of DMACH 2011 (2011)
34. Li, Z., Zhu, Q., Gold, C.: Digital Terrain Modeling: Principles and Methodology. CRC press, Boca Raton (2010)
35. Longay, S., Runions, A., Boudon, F., Prusinkiewicz, P.: Treesketch: interactive procedural modeling of trees on a tablet. In: Proceedings of the international symposium on sketch-based interfaces and modeling, pp. 107–120. Eurographics Association (2012)
36. Loop, C.: Smooth subdivision surfaces based on triangles. Ph.D. thesis (1987)
37. Losasso, F., Hoppe, H.: Geometry clipmaps: terrain rendering using nested regular grids. ACM Trans. Graph. (TOG), **23**, 769–776 (2004)
38. Mahdavi-Amiri, A., Harrison, E., Samavati, F.: Hexagonal connectivity maps for digital earth. Int. J. Digital Earth 1–20 (2014)
39. Maidment, D.: Arc Hydro: GIS for Water Resources. No. v. 1 in Arc Hydro: GIS for Water Resources. ESRI Press (2002). https://books.google.ca/books?id=07vH7Sf0v6MC
40. Musialski, P., Wonka, P., Aliaga, D.G., Wimmer, M., Gool, L., Purgathofer, W.: A survey of urban reconstruction. In: Computer Graphics Forum, vol. **32**, pp. 146–177. Wiley Online Library (2013)
41. Olsen, L., Samavati, F., Jorge, J.: Naturasketch: modeling from images and natural sketches. IEEE Comp. Graph. Appl. **31**(6), 24–34 (2011)
42. Olsen, L., Samavati, F.F.: Image-assisted modeling from sketches. In: Proceedings of Graphics Interface 2010, GI '10, pp. 225–232 (2010)
43. Olsen, L., Samavati, F.F., Sousa, M.C., Jorge, J.A.: Sketch-based modeling: A survey. Comp. Graph. **33**(1), 85–103 (2009)
44. Pakdel, H.R., Samavati, F.: Incremental adaptive loop subdivision. In: Computational Science and Its Applications–ICCSA 2004, pp. 237–246. Springer (2004)
45. dos Passos, V.A., Igarashi, T.: Landsketch: A first person point-of-view example-based terrain modeling approach. In: Proceedings of the International Symposium on Sketch-Based Interfaces and Modeling, pp. 61–68. ACM (2013)
46. Poullis, C., You, S.: Photorealistic large-scale urban city model reconstruction. IEEE Trans. Visual. Comp. Graph. **15**(4), 654–669 (2009)
47. Pusch, R., Samavati, F.: Local constraint-based general surface deformation. Shape Model. Int. Conf. (SMI) **2010**, 256–260 (2010)
48. Santoro, F., Tarantino, E., Figorito, B., Gualano, S., D'Onghia, A.M.: A tree counting algorithm for precision agriculture tasks. Int. J. Digital Earth **6**(1), 94–102 (2013)
49. Schmidt, R., Khan, A., Kurtenbach, G., Singh, K.: On expert performance in 3d curve-drawing tasks. In: Proceedings of the 6th eurographics symposium on sketch-based interfaces and modeling, pp. 133–140. ACM (2009)
50. Severn, A., Samavati, F., Sousa, M.C.: Transformation strokes. In: Proceedings of the Third Eurographics Workshop on Sketch-Based Interfaces and Modeling, SBM'06, pp. 75–81 (2006)
51. Simakov, D., Caspi, Y., Shechtman, E., Irani, M.: Summarizing visual data using bidirectional similarity. In: Computer Vision and Pattern Recognition, 2008. CVPR 2008., pp. 1–8. IEEE (2008)
52. Sinha, S.N., Steedly, D., Szeliski, R., Agrawala, M., Pollefeys, M.: Interactive 3d architectural modeling from unordered photo collections. ACM Trans. Graph. (TOG), **27**, 159 (2008)
53. Smelik, R.M., Tutenel, T., Bidarra, R., Benes, B.: A survey on procedural modelling for virtual worlds. pp. 31–50 (2014)
54. Stachniak, S., Stuerzlinger, W.: An algorithm for automated fractal terrain deformation. Comp. Graph. Artif. Intel. **1**, 64–76 (2005)
55. Tan, P., Fang, T., Xiao, J., Zhao, P., Quan, L.: Single image tree modeling. ACM Trans. Graph. (TOG), **27**, 108 (2008)
56. Tasse, F.P., Emilien, A., Cani, M.P., Hahmann, S., Dodgson, N.: Feature-based terrain editing from complex sketches. Comp. Graph. **45**, 101–115 (2014)
57. Wang, W., Jüttler, B., Zheng, D., Liu, Y.: Computation of rotation minimizing frames. ACM Trans. Graph. (TOG) **27**(1), 2 (2008)
58. Xiao, J., Fang, T., Zhao, P., Lhuillier, M., Quan, L.: Image-based street-side city modeling. ACM Trans. Graph. (TOG) **28**(5), 114 (2009)

59. Xue, T., Liu, J., Tang, X.: 3-d modeling from a single view of a symmetric object. IEEE Trans. Image Process. **21**(9), 4180–4189 (2012)
60. Yang, L., Wu, X., Praun, E., Ma, X.: Tree detection from aerial imagery. In: Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 131–137. ACM (2009)
61. Zhou, H., Sun, J., Turk, G., Rehg, J.M.: Terrain synthesis from digital elevation models. IEEE Trans. Visual. Comp. Graph. **13**(4), 834–848 (2007)
62. Zlatanova, S., Rahman, A.A., Shi, W.: Topological models and frameworks for 3d spatial objects. Comp. Geosci. **30**(4), 419–428 (2004)
63. Zorin, D., Schröder, P., Sweldens, W.: Interactive multiresolution mesh editing. In: Proceedings of the 24th annual conference on Computer graphics and interactive techniques, pp. 259–268. ACM Press/Addison-Wesley Publishing Co. (1997)

Faramarz Samavati is a full professor of the Department of Computer Science at the University of Calgary. His research interests include computer graphics, visualization, 3D imaging and geometric modeling. Dr. Samavati has published more than hundred papers, one book and filed 3 patents. In the past four years, he has received four best paper awards, Digital Alberta Award, Great Supervisor Award, and University of Calgary Award, which honors his contribution to the development of new technologies and innovations.

Adam Runions is a Postdoctoral researcher at the Max Planck Institute for Plant Breeding Research. He is a part of the plant development and diversity laboratory of the Department of Comparative Development and Genetics. His research interests are computational modeling of plant form and development, and geometric modeling for computer graphics and computer-aided design. Previously, he worked in the Biological Modeling and Visualization research group and Graphics Jungle Computer Graphics lab of the Computer Science department at the University of Calgary. There he completed a PhD under the supervision of Przemyslaw Prusinkiewicz and Faramarz Samavati.