TECHNIQUES FOR AUTOMATIC CREATION OF TERRAIN DATABASES FOR TRAINING AND MISSION PREPARATION

Frido KUIJPER, Rob VAN SON, Frank VAN MEURS, Ruben SMELIK, Klaas Jan DE KRAKER
TNO Defence, Security and Safety
The Hague, The Netherlands

ABSTRACT

In the support of defense agencies and civil authorities TNO runs a research program that strives after automatic generation of terrain databases for a variety of simulation applications. Earlier papers by TNO at the IMAGE conference have reported in-depth on specific projects within this program. Rather than being in-depth on a specific subject, this paper provides a more global update on recent results that were obtained in a number of different projects. The projects have focused on techniques to create geo-specific terrain databases from imagery and on techniques that procedurally create geo-typical terrain based on sketch input. The results demonstrate that automatic techniques are available for the creation of terrain databases that are fit-for-purpose for most training and part of the mission rehearsal simulation applications.

INTRODUCTION

The effectiveness of simulation applications for training and mission rehearsal is greatly influenced by the availability of high quality terrain databases. The creation of these databases is typically performed in three possible ways:

- The terrain is automatically built using terrain generation software. The input data consists of externally acquired GIS data that is readily available: elevation data, imagery and vector data.
- The terrain is automatically built using terrain generation software, but only elevation data and imagery are acquired externally from readily available sources. The vector data describing the features in the terrain is generated by manual editing using the imagery as input.
- The terrain is fully manually modeled using an interactive 3D modeling tool. This method is often applied for small terrains, with a high level of detail. Either real world maps/images or imaginary maps/sketches are used as input.

The latter two methods will normally generate detailed results, but at the cost of significant manual labor. The first method is more attractive in terms of the amount of manual editing that is required. However, three main problems arise when working from readily available GIS data:

- When the GIS data is acquired from various sources, correlation errors are likely to occur.
- For remote locations, these data sources will be either not available or of poor quality.
- The data will typically not allow for accurate 3D modeling of features.

To overcome these problems while still minimizing the amount of manual editing, automatic techniques are needed to extract the required GIS data from sensor data sources. The first main subject of this paper addresses these techniques, working from imagery as the primary data source.

The second main subject treats automatic terrain creation from a different perspective. Building terrain databases automatically from geo-specific source data can be very efficient but, in some cases, does not deliver the most effective database for the purpose of the simulation. For mission rehearsal and training exercises with live components involved, the use of geo-specific source data is mandatory since the terrain database should accurately resemble the real mission area for these cases. Often, the same type of geo-specific database is also used for more basic training purposes. Given a specific training task, the geo-specific terrain is searched for a location that is suitable for a scenario serving this particular training task. This can be a valid approach, since building a terrain database from geo-specific data can be cheaper than fully manually modeling a terrain that fits the purpose. However, if better automatic techniques were available that create an imaginary terrain that fits the training purpose, this would result in more effective terrain databases at lower cost. In this paper, we report on techniques that pursue that goal with SketchaWorld, a concept that creates detailed terrain databases using procedural techniques based on sketch user input.

Presented at the **IMAGE 2010** Conference
Scottsdale, Arizona – July 2010

AUTOMATIC MODELING FROM IMAGERY

The challenge for automatic modeling from imagery is to build detailed models straight from image source data with a minimum of manual editing. For remote locations, the process usually starts off with satellite imagery, gradually moving into better images that are aerial or ground acquired. The modeling techniques should be aiming at getting the most out of each image source as well as fusing the results into a coherent model.

The actual value of the resulting model will depend on its purpose: training, mission rehearsal, LVC scenarios and decision support all have their specific demands for accuracy.

Modeling from Satellite Imagery

Modern stereo imagery satellites - while increasing their resolution, accuracy and versatility - are becoming more and more effective for terrain modeling. In this paper we report on results we have achieved with a Quickbird stereo image pair.

Case study setup

We worked on modeling a 15 km x 15 km part of northern Oman around the city of Khasab. Two Quickbird images were acquired for this area. The images (see Fig. 1) provide a 60 cm resolution panchromatic band and four 2.4 meter resolution multispectral bands. The two images were taken 4 months separated in time.

All the processing was done with COTS software from ERDAS. Subsequent simulation databases were generated with Presagis TerraVista and post processed for TNO's proprietary EVE simulation platform (which has an OpenSceneGraph based visual system).

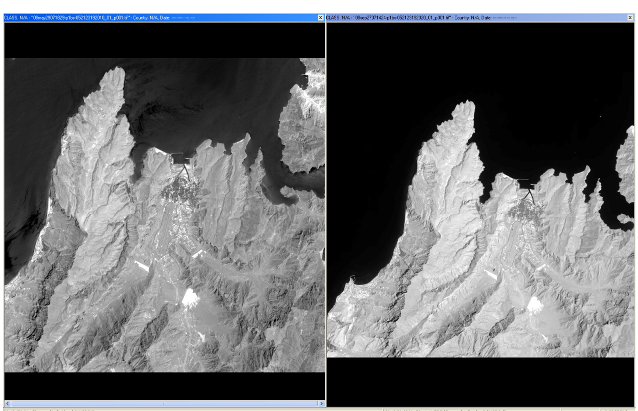


Fig. 1: Quickbird stereo pair images for Khasab area (here showing only the panchromatic channel). Approximate size of the images is 15 km x 15 km.

Results

From the imagery, we modelled the terrain skin as a regular gridded Digital Terrain Model (DTM, modeling the bare earth skin), orthorectified the imagery and extracted several terrain features from the ortho imagery:

- coastline;
- vegetation;
- buildings (footprint only);
- roads.

Terrain skin model. The Quickbird imagery used in this experiment is not the most suited for work on terrain skin modeling using photogrammetric techniques. The images are not taken on the same moment in time, but 4 months separated in time. Nevertheless, we were able to obtain an acceptable DTM with a resolution of 6 meters. The automatic feature filtering of the ERDAS LPS photogrammetry suite was used to generate the DTM. However, manual editing of the DTM is still required to improve breaklines (e.g. on the road that follows the steep coastline) and to remove elevated features (buildings, vegetation) interfering with the DTM.

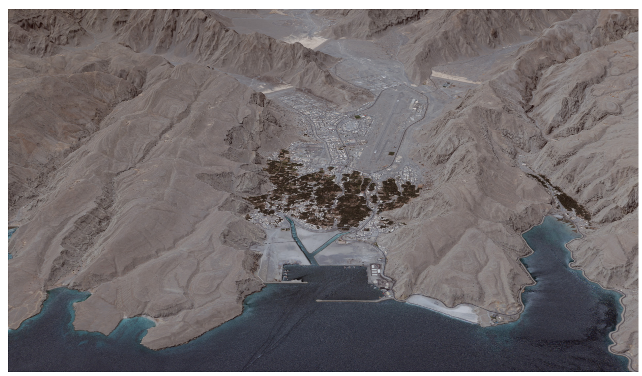


Fig. 2: An impression of the DTM with draped ortho-imagery.

Coastlines. In order to support sea surface and underwater scenarios, we needed to identify coastline vectors that are accurate up to the image resolution. We generated these detailed vectors – as we did for all the feature extraction in this project – by using the ERDAS Imagine Objective tool. Objective allows for training feature characteristics in both the pixel as well as the vector domain. From the training samples, a Bayesian stochastic network is created that allows for pixel and vector classification [4]. For the coastline, the algorithm was trained with manually selected samples of sea parts and non-sea parts using only spectral properties. Practical limitations of the tool refrained this from being a one-click operation. The coastline was generated in parts,

Presented at the **IMAGE 2010** Conference
Scottsdale, Arizona – July 2010

with locally optimized training and merging of the resulting coastline vectors in a post-process stage.

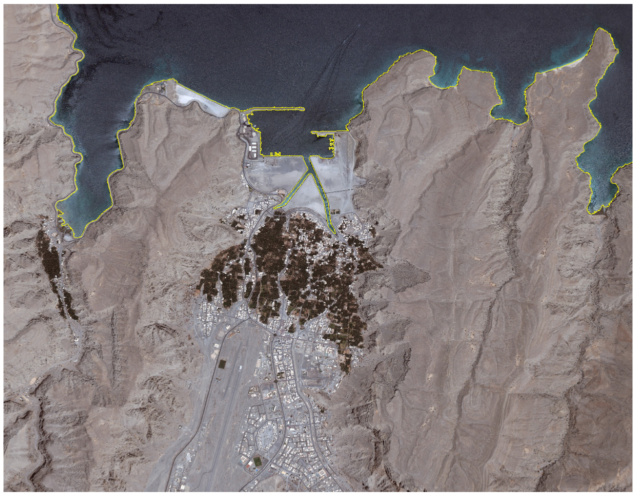


Fig. 3: Accurate coastline vectors are created through image segmentation and spectral classification.

Vegetation. For the extraction of vegetation features we identified two main classes of trees present in the subject terrain: palm trees and deciduous trees. The classification was done through segmentation, spectral matching and vectorisation and yields good results as illustrated in Fig. 4 (a).

Buildings. For buildings, the footprint vectors were extracted using both pixel characteristics as well as vector characteristics. Typical size and shapes of buildings help to determine whether a specific feature is a building or not, since spectral properties do not always allow to discriminate between terrain and buildings. In Fig. 4 (b) it can be seen that the resulting vectors provide a fairly good solution. The vectors that we find are accurately matching the building they represent. We do see approximately 22% missed features (features that are in the image, but not extracted) and 3% false features (features that are extracted but that are not present in the image). It is our impression that the error rate in missed features could be improved by identifying more accurate classes of building types and put more effort in choosing the training samples. At the same time, this indicates that the present workflow will always ask for strict quality control if results need to be accurate.

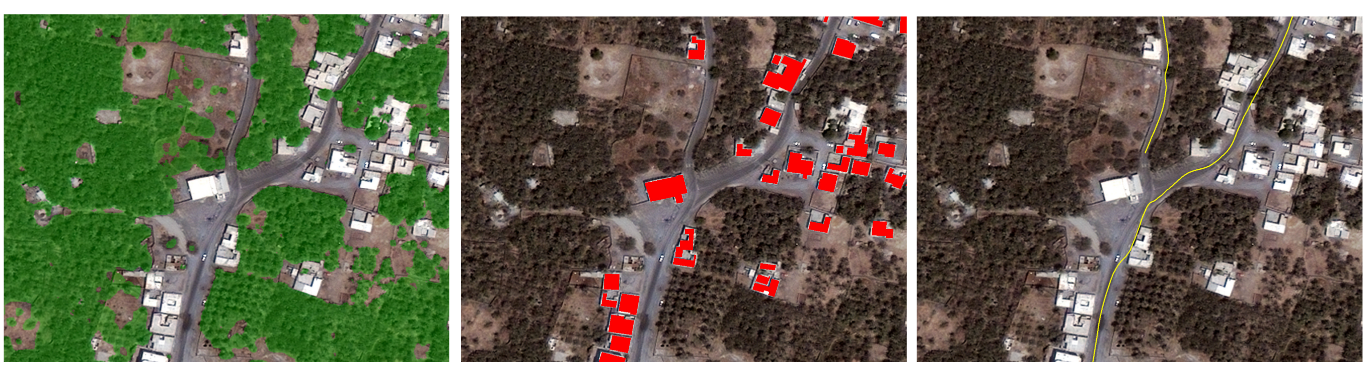
In the final database, the building footprints were populated with a geo-typical, textured building representation with non-specific visual appearance and building height (see Fig. 6).

Roads. Extraction of road line vectors is more difficult than it seems at first sight. Roads are very predominant features to the human eye, but are very difficult to detect by a computer. Fig. 5 illustrates the algorithm that was used to extract roads. Similar to the other features, the image is segmented and assigned pixel based probabilities. After selecting the likely road segments, the resulting pixels are converted to line vector data. Rather than the smooth sample in Fig. 5, Fig. 4 (c) shows a more hard practical sample where spectral properties are not so clean due to trees and cars occluding the road.

Discussion

The terrain database that was built for the Khasab area on the basis of automatic processing of satellite imagery has an appealing impression, but the result is not obtained without pain, nor is it perfect.

Terrain skin modeling still needs manual editing to properly filter features from the bare earth. Automatic filtering could be improved by integrating the feature classification with feature filtering for the terrain.



(a) **(b)** **(c)**

Fig. 4: Automatic feature extraction to generate (a) vegetation, (b) building footprints and (c) road vectors.

The feature extraction algorithms used by ERDAS based on learning algorithms that consider object based pixel and vector properties are a good way to go, but the algorithms are hard to setup effectively and suffer from practical limitations to be as productive as desired.

Bottom line is that the techniques used in this experiment are a valid approach to automatically building terrain databases, as long as the applications purpose allows for an 80% solution. For training, this will be valid in most cases. For mission rehearsal it can be sufficient, if only a rough awareness of the mission area is needed. As soon as details in the terrain and its infrastructure start to count, e.g. in operational decision support or LVC scenario's, the databases will need more manual adjustment.

Note that these results do not demonstrate the full state-of-the-art, since better results can be achieved with the latest satellites that have higher resolution, broader spectral coverage and in-track stereo capabilities. At the time this experiment was started, QuickBird imagery was the best available data, while currently DigitalGlobe and GeoEye would offer better data with the WorldView-2 and GeoEye-1 satellites.

Fig. 6: Resulting database shown in visual system.

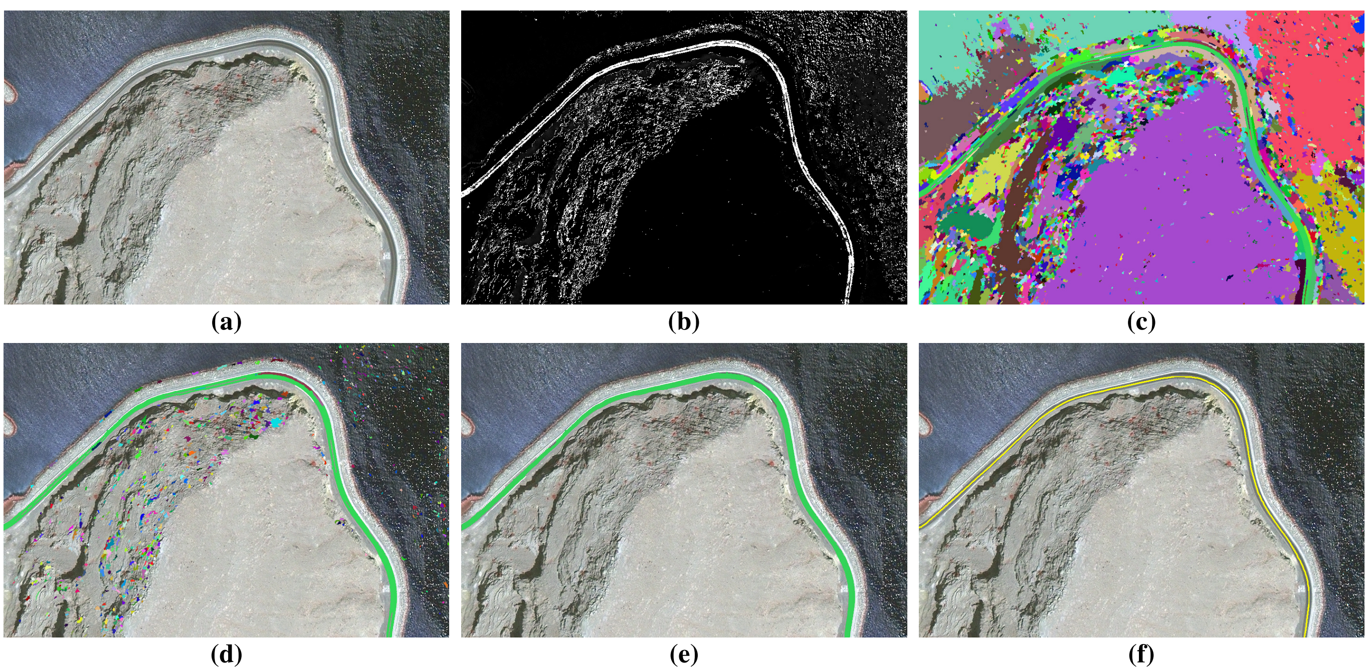


Fig. 5: Road feature vector extraction: (a) input image, (b) pixel probability computation, (c) segmentation with per segment probability computation, (d) probability filtering of segments, (e) size filtering and (f) vectorization.

Modeling from Short Range Imagery

The results from the previous paragraph look appealing, but they fall short in the accurate representation of the buildings in the environment. As presented earlier at Image [3] we stress the importance of short range imagery, preferably from ground based sensors. Short range imagery is required to build accurate building geometry and spectral representation. We have not come to a full solution for this problem, but work in progress focuses on specific elements of the chain that works from short range imagery towards 3D models.

Bundle adjustment pipeline

Typically, the 3D reconstruction of an object from multiple view imagery consists of a pipeline that has the following steps:

- find matching keypoints between images, i.e. points that represent the same real world position in different images;
- perform bundle adjustment by estimating the sensor model parameters and camera positions, while solving the constraints imposed by the matching keypoints;
- extract a point cloud from the adjusted and registered images by matching as many pixels as possible;
- match a 3D model to the point cloud.

Keypoint detectors are key

Obviously, the keypoint detectors and matching algorithms used in the bundle adjustment pipeline play a cardinal role. In the case of short range imagery, keypoint detection and matching is significantly more difficult than is the case for satellite imagery. The main reasons for this are:

- The large relative camera position change (relative to the distance between camera and scene) induces more perspective distortion of features.
- More motion parallax is induced, potentially causing significant changes in the image around a keypoint.

Research direction for improved keypoint detector

Current research in our program focuses on the improvement of keypoint detector performance for short range imagery. One algorithm which is commonly used for the detection and representation of keypoints is the Scale-Invariant Feature Transform (SIFT) [4]. It has been designed to be invariant to changes in orientation, scale and illumination and has even been shown to be robust against affine changes induced by small changes in viewpoint. SIFT has been empirically shown to outperform a number of other current methods for representing keypoints [6]. However, in cases of significant changes in viewpoint the performance of SIFT in matching keypoints between images drops significantly [4].

We investigated the performance of SIFT on a set of aerial images of an urban environment (see Fig. 7). These were acquired from a helicopter equipped with two cameras, one of which faced forwards to obtain oblique imagery and one of which faced downwards to obtain near nadir imagery. The images subsequently served as input to Bundler [16], a freely available off-the-shelf 3-dimensional reconstruction package, which extracts SIFT keypoints and subsequently performs bundle adjustment [17], [1], in order to create a 3-dimensional point cloud.

The results of this investigation showed several characteristics with regard to matches between images:

- keypoints from consecutive forward images can be matched;
- keypoints from downward images can be matched;
- keypoints from non-consecutive forward images can not be matched;
- keypoints between forward and downward images can not be matched.

Fig. 7: A few sample images from the dataset, acquired with a forward and downward camera on a helicopter. Note the large perspective differences while looking at the same objects.

Or, concisely put, images separated by a large change in viewpoint can not be matched, as predicted. As a result, a very large part of the information which is present in the image dataset with regard to the 3-dimensional structure of the original environment is simply unused, resulting in errors and an unnecessarily low density in the created point-cloud.

Although SIFT is shown to be unsuitable in case of large viewpoint changes, its other characteristics, i.e. its invariance to rotation, scale and illumination and its performance with regard to unambiguous keypoint representation, make it a very well performing algorithm in general. As a consequence, there have been attempts at improving on the initial algorithm in order to tackle the case of large viewpoint changes, e.g. PCA-SIFT [2] and ASIFT [7], which show better performance results. Although both methods show better performance, none provide explanations as to why SIFT itself shows such poor performance under the given conditions.

Our current research at TNO is aimed at answering exactly this question. The large changes in camera position induce both large perspective distortion of the image and its features as well as motion parallax which results in abrupt image changes through occlusion. To what extent this exactly affects the keypoints and their representation is still unknown, as well as to what extent these characteristics remain constant. In answering these questions, we hope to identify the specific aspects of the algorithm which induce the problematic characteristics, so that these may be improved upon.

Further, not only does matching of keypoints require similarity between candidates, it also requires a suitable means of comparing keypoint candidates. As such, the second part of our research is aimed at the evaluation of existing schemes and the development of novel ones for matching keypoint candidates. Previous research by Pele and Werman [8] has already shown that the comparison method for SIFT keypoints proposed by Lowe [4] can be improved upon, e.g. through their proposed Earth Mover's Distance variant. Also, other distance measures and matching schemes can be conceived of which are better suited for matching histogram-based representations.

Taken together, by developing such improved keypoint detection and representation methods and methods for comparing keypoints candidates, we aim to allow for matching of not only images with minor changes, but also of images under large viewpoint changes. As a consequence, the 3D reconstruction pipeline will become more robust under datasets that are acquired from ground based and aerial platforms. The requirements for image overlap can be lowered and thus the cost of data acquisition will be reduced.

AUTOMATIC CREATION OF GEO-TYPICAL TERRAIN

In the modeling and simulation community, emphasis has typically been put on creating geo-specific terrain models, thereby somewhat neglecting the alternative: geo-typical terrain. The main advantage of geo-typical terrain is the absence of any constraints imposed by real-world correlation. Such a terrain model can be fit exactly to the scenario requirements and possible learning objectives. This makes geo-typical terrain modeling especially suitable for training scenarios, for which it might result in a superior learning experience.

For geo-typical terrain creation, the modeling challenge changes from the acquisition and processing of geo-data to designing a fictional virtual world from scratch. This requires not only some amount of inspiration and creativity, but also a substantial amount of modeling effort. Due to time and budget constraints, modeling a virtual world completely by hand, as done in the entertainment game industry, is almost always not an option. This means that automatic modeling techniques have to be applied to generate at least parts of the landscape.

These content generation techniques, collectively known as procedural methods, can automatically generate results which are in some cases very close to the quality of hand-modeled content. However, as discussed in [10], they currently have at least three limitations which hinder their effective application to geo-typical terrain modeling:

- they can be quite unintuitive in use, forcing end-users to set input parameters that require in-depth knowledge of their internal workings;
- they offer very limited user control, resulting in output that is too random to be practical;
- because they focus on generating one specific terrain feature, automatically integrating all these features to form a believable world remains an open issue.

The SketchaWorld approach

The goal of the SketchaWorld project [15], a combined research effort of TNO with Delft University of Technology, is to provide designers with an efficient, automatic virtual world creation method that allows for control over the global layout as well as more detailed properties of the landscape. Furthermore, it strives to reduce the complexity of virtual world modeling and procedural methods, and be accessible for non-specialist users, such as training instructors, etc. [14]. Using an easy to use input method called procedural sketching, designers can specify their virtual world, which is interactively created while they sketch it out. Once satisfied with the generated results, designers can save the 3D virtual world model and export the model to other relevant formats, such as GIS data.

Procedural sketching
Procedural sketching provides two interaction modes:

- **Landscape mode** Designers paint a top view of the landscape by coloring a grid with ecotopes (a small area of homogeneous terrain and features). These ecotopes encompass both elevation information (elevation ranges, terrain roughness) and soil material information (sand, grass, rock, etc.). The grid size is adjustable and the brushes used are very similar to typical brushes found in image editing software, including draw, fill, lasso, magic wand, etc.
- **Feature mode** Designers place elements like rivers, roads, and cities on the landscape using vector lines and polygon tools. This resembles the basic tools found in vector drawing software: placing and modifying lines and polygons is done by manipulating control points.

To directly see the effect of edit actions on the virtual world model, users sketch on top of a 2D view of the generated terrain layers. Each sketch element is procedurally expanded to a matching terrain feature. The 2D top view and a 3D preview are both updated immediately as new results are generated. Depending on the interaction mode, an overlay is displayed representing relevant elements of the user sketch. Fig. 8 shows the grid overlay that assists with painting the landscape (top left) and shows the line drawing overlay for displacing a river (bottom left).

An important feature of the framework is the support for iterative modeling, which makes the workflow more closely resemble the way designers work with traditional manual modeling systems. An iterative workflow requires:

- a short feedback loop between edit action and the visualization of its effects;
- an edit history and unlimited undo and redo capabilities;
- clear and fast updating different views on the model being edited.

To obtain a short feedback loop, each edit action should be executed interactively and the effects of this action should be displayed immediately. Because we cannot always guarantee interactive updates, especially not for operations affecting large areas, we implemented an asynchronous execution setup, in which designers can continue to sketch without interruption. Furthermore, designers are presented with an edit history with which they can step through a list of all their editing actions, undoing or redoing one or more actions. The design and implementation of the iterative workflow for interactive procedural sketching are discussed in [11].

Procedural generation
Once a designer has performed an edit action (e.g. painted part of the landscape or defined a terrain feature's coarse outline), a custom procedure generates the appropriate content. These procedures are based on well known procedural methods (see e.g. [10]).
For the landscape, the painted grid of ecotopes steers the generation process. Each ecotope defines, among other things, elevation ranges, terrain roughness and type of noise (e.g. smooth, hilly, rocky, erosive). The procedure combines standard fractal noise-based terrain generation with interpolation of the coarse grid. Ecotopes also are a factor in, for instance, determining the suitability of a certain area for a specific species of trees.
For each terrain feature, a specific procedural method is used

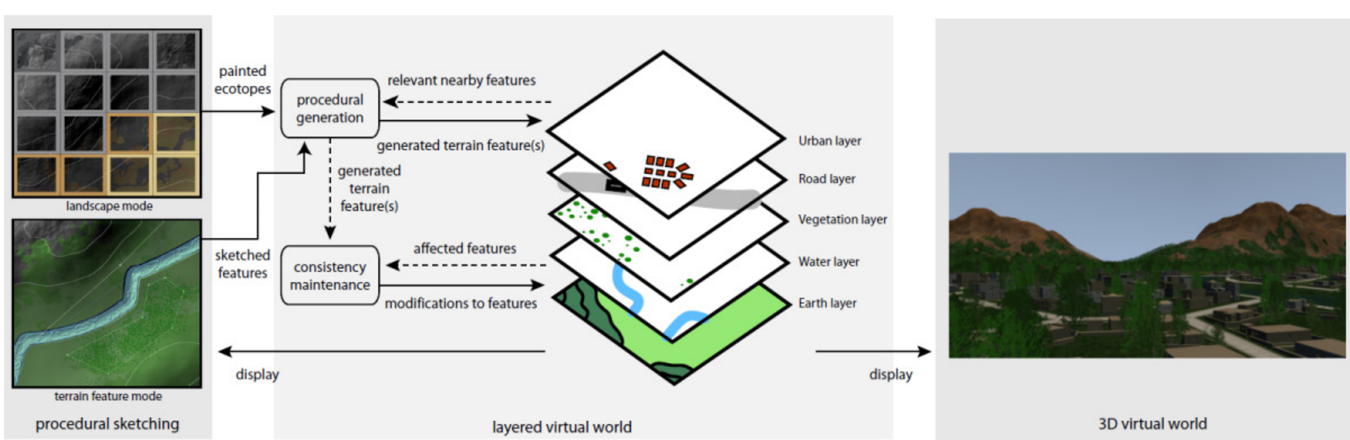


Fig. 8: Overview of the workflow of SketchaWorld.

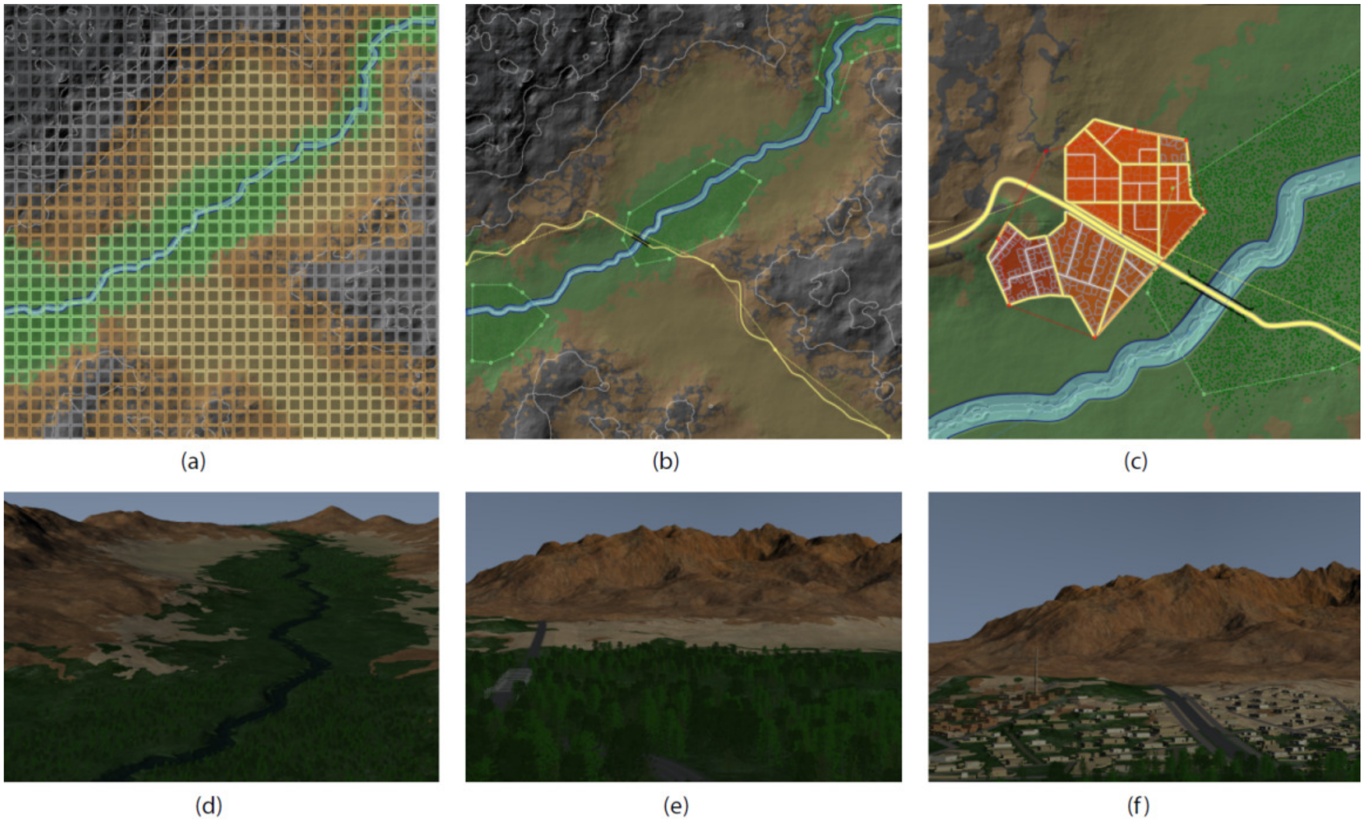


Fig. 9: Example procedural sketching session, resulting in a geo-typical terrain database [12].

that follows these outlines to a certain extent, but of course generates a more detailed structure, taking into account the landscape and nearby other features. Typically, these procedures have several phases, in which the coarse outline is first refined to a concrete structure of the feature and later detailed with individual objects (trees, buildings, etc.).

Consistency maintenance

All generated terrain features are grouped in five logical layers, inspired from GIS (see Fig. 8 middle). Terrain features need to blend in with their surroundings to form a lifelike virtual world. If these features were to be generated separately from their context in the virtual world, designers would have to perform their integration manually, which would harm their productivity and limit their ability to experiment. Because the layered virtual world model is semantically rich, it can be automatically kept in a valid state using a form of consistency maintenance.

The introduction of a new terrain feature into the layered virtual world model discerns two phases (shown in Fig. 8):

- The feature's path, shape and other properties are determined based on the provided user parameters and on relevant nearby existing terrain features;
- Some of the surrounding features are affected by the newly introduced feature and are, for instance, connected to it, modified in some way or even removed.

Consistency maintenance in these two phases is based on a set of rules, describing the mutual influence of terrain features. Although these maintenance phases somewhat increase the execution time of individual operations, this continuously keeps the virtual world model in a consistent and usable state and it allows the designer to quickly see possible local side-effects of edit actions. Each time a terrain feature is modified, changes to related features are performed automatically as logical side-effects of this change.

Results

At IMAGE 2009, we have shown how one can customize existing procedural methods to generate terrain features in a style typical to South Afghanistan [13]. The new contributions of this research are the interactive procedural sketching facilities and the consistency maintenance mechanisms. These give designers not only more direct feedback, but also a finer level of user control, which allows them to more effectively steer the automated procedures to generate a desired end result.

We present a short example sketch session (in Fig. 9), to give an impression of how one would work with procedural sketching. In this example, a designer creates a natural landscape with a river flowing through a valley and a city on a hill along this river. Fig. 9.a shows the basic landscape, sketched in *landscape mode*: a green valley encompassed by mountains, with some forests defined in the valley and a river flowing through it (see also Fig. 9.d). On top of this natural environment, in *feature mode*, some man-made features are added. Firstly, the designer coarsely outlines the desired path of a major road Fig. 9.b. This road crosses the river at one point, Fig. 9.e shows the bridge that was automatically created. Lastly, the designers outlines a small village along this road (Fig. 9.c), at the hillside (Fig. 9.e).

Challenges ahead

With procedural sketching, designers can quickly obtain a virtual world that matches their requirements on the level of terrain features and their relations and connections. However, on the more detailed level of individual objects, designers will often want to manually edit and tune the generated results to fit more precisely to their requirements, which is currently not possible. The challenges of integrating manual edit facilities with procedural generation methods are discussed in [12]. We believe that this combination will provide designers with the productivity gain of procedural methods, while still allowing for a fine level of user control and flexibility.

Furthermore, we would like to address the generation of larger-scale landscapes. The main challenge will be how to interactively model such environments, as this poses all kinds of technical issues related to performance, efficiency, level of detail, and database file sizes.

Lastly, in the coming year we plan to evaluate both the user experience of procedural sketching and the suitability of the generated output for serious games and simulations.

FUTURE DIRECTIONS

Our research will stay focused on finding improved techniques for automatic terrain modeling from imagery for mission specific environments. Although laser measurements are more and more used in this process, a significant role for imagery will persist. The development of algorithms for laser-point-cloud-like measurements from imagery will strenghten this role. Research directions include the integration of feature extraction and feature filtering for DTM terrain skin modeling, as well as the integration of uncalibrated, unregistered short range imagery in the process.

Another direction we would like to explore is the combination of geo-specific GIS data with procedural modeling. For most areas of the world, some source data is available for reasonable prices, but the available source data might lack the level of detail required to obtain a terrain model, usable for a specific simulation purpose. This means that additional modeling is often required, either by augmenting the source data (for instance, automatically or manually interpreting and extracting features from a satellite image as described in this paper) or by enriching the 3D terrain model using standard 3D modeling techniques. Procedural modeling would seem like an ideal method to enhance the coarse data with finer structure and details if full geo-specific modeling is not required. Most of the current commercial packages can add procedural detail to a limited extent by providing, for instance, a method to randomly scatter objects in a polygonal area. However, a noteworthy exception is PixelActive's CityScape [9], which allows for hybrid of manual and procedural modeling of urban environments, although it provides a somewhat limited and narrowly focused set of procedural operations. We think it would be beneficial to apply more advanced procedural methods to enrich coarse GIS data.

CONCLUSIONS

In this paper we have presented several techniques that provide automatic modeling of terrain databases, be it geo-specific from imagery or geo-typical from quick sketching and procedural modeling. In both cases we believe that the resulting models are fit-for-purpose for many training and even part of the mission rehearsal applications.

For training applications, the procedurally generated models of the SketchaWorld concept are often more effective than geo-specific models, since the content is based on scenario requirements and every part of the model is completely accurately known (facilitating e.g. sensor simulation).

Geo-specific models that are automatically derived from imagery offer an '80% solution' out of the box. Significant manual editing of the results is required as soon as requirements for accuracy and detail increase, e.g. in the case of critical decision support systems and LVC scenario's with combined live and virtual/constructive simulations. Future research will lower the threshold, but it will take many more years to reach the ultimate automated method that delivers accurate geometry along with semantically rich classification of the terrain and its features.

ACKNOWLEDGMENTS

This research has been supported by the Dutch Ministry of Defense as well as the GATE project, funded by the Netherlands Organization for Scientific Research (NWO) and the Netherlands ICT Research and Innovation Authority (ICT Regie).

AUTHOR BIOGRAPHIES

Frido Kuijper is a senior research scientist at TNO Defence, Security and Safety, a Dutch research institute. He holds a Master's degree in Computer Graphics from Delft University of Technology and has 18 years experience in the field of Modeling & Simulation for military applications. Previous work of Frido has been in the field of networked simulation architectures and virtual environment technology. Currently, he is the lead scientist for TNO in research that aims at the generation and application of synthetic natural environments for Defence and Civil Security applications.

Rob van Son is a research scientist for TNO Defence, Safety and Security in The Hague, The Netherlands. Currently, he's performing research within TNO's modeling and simulation research program on synthetic environment modeling techniques, focusing on synthetic natural environment generation and representation. He holds a master's degree in Computer Science from Utrecht University.

Frank van Meurs is a research scientist for TNO Defence, Security and Safety in The Netherlands. He is a PhD candidate on a project entitled "Automatic World Generation Based on Real Data", in close cooperation with Utrecht University. This project aims at developing methods for automatically generating virtual environments of real-world scenes. The project is part of the Dutch research program "Game Research for Training and Entertainment (GATE)".

Ruben Smelik is a research scientist for TNO Defence, Security and Safety in The Netherlands. He is a PhD candidate on a project entitled "Automatic Creation of Virtual Worlds", in close cooperation with Delft University of Technology. This project aims at developing new tools and techniques for creating geo-typical virtual worlds for serious games and simulations. The project is part of the Dutch research program "Game Research for Training and Entertainment (GATE)". Ruben holds a master's degree in Computer Science from the University of Twente.

Klaas Jan de Kraker is a member of the scientific staff at TNO Defence, Security and Safety. He holds a Ph.D. in Computer Science from Delft University of Technology. He has a background in Computer-Aided Design and Manufacturing, collaboration applications, software engineering (methodologies), meta-modeling and data modeling. Currently he is leading various simulation projects in the areas of simulation based performance assessment, collective mission simulation, multifunctional simulation and serious gaming.

REFERENCES

[1] Hartley, R. and Zisserman, A., 2003, *Multiple View Geometry in Computer Vision,* Cambridge University Press New York, NY, USA, 123-127.

[2] Ke, Y and Sukthankar, R., 2004, "PCA-SIFT: A More Distinctive Representation for Local Image Descriptors", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* IEEE, vol.2, 506-513

[3] Kuijper, F., Son, R. van and Veraart, M., 2008, "Urban Environment Modeling from Mixed Airborne and Ground-Based Imagery", *Proceedings of the Image 2008 Conference.*

[4] Lane, M.A., Anderson, I. and Kloer, B., 2009, "Multi-class Automated and semi-automated feature extraction using ERDAS IMAGINE Objective". *Proceedings of ESA-EUSC-JRC 2009: Image Information Mining: automation of geospatial intelligence from Earth Observation, nov 3-5 2009, Madrid, Spain.*

[5] Lowe, D. G., 2004. “Distinctive Image Features From Scale-Invariant Keypoints”. *International Journal of Computer Vision IJCV*. vol. 60, 91-110.

[6] Mikolajczyk, K. and Schmid, C, 2005, "A Performance Evaluation of Local Descriptors", *IEEE Transactions on Pattern Analysis and Machine Inteligence*, IEEE, vol.27, 1615-1630

[7] Morel, J.M. and Yu, G, 2009, "ASIFT: A New Framework for fully Affine Invariant Image Comparison", *SIAM Journal on Imaging Sciences SIIMS.* vol. 2, 438-469.

[8] Pele, O. and Werman, M., 2008, "A Linear Time Histogram Metric for Improved SIFT Matching", *Proceedings of the 10th European Conference on Computer Vision: Part III*, INRIA Grenoble – Rhône-Alpes, vol.1, 495-508

[9] PixelActive, 2010, CityScape 1.8, available from http://pixelactive3d.com/Products/CityScape

[10] Smelik, R. M., de Kraker, K. J., Tutenel, T., Bidarra, R., and Groenewegen, S. A., 2009. A Survey of Procedural Methods for Terrain Modeling. *Proceedings of the CASA Workshop on 3D Advanced Media In Gaming And Simulation (3AMIGAS)*, Amsterdam, The Netherlands.

[11] Smelik, R. M., Tutenel, T., de Kraker, K. J., and Bidarra, R., 2010. "Interactive Creation of Virtual Worlds Using Procedural Sketching", *Proceedings of Eurographics 2010 - Short papers,* Norrköping, Sweden.

[12] Smelik, R. M., Tutenel, T., de Kraker, K. J., and Bidarra, R., 2010. Integrating procedural generation and manual editing of virtual worlds. In *Proceedings of the FDG 2010 Workshop on Procedural Content Generation in Games*, Monterey, CA, USA.

[13] Smelik, R., de Kraker, K. J., Tutenel, T., and Bidarra, R., 2010, "A Case Study on Procedural Modeling of Geo-Typical Southern Afghanistan Terrain", *Proceedings of the Image 2009 Conference.*

[14] Smelik, R., de Kraker, K. J., Tutenel, T., and Bidarra, R., 2010. Declarative Terrain Modeling for Military Training Games. *Int. Journal of Computer Game Technology*.

[15] Smelik, R., Tutenel, T., de Kraker, K. J., and Bidarra, R., 2008. A Proposal for a Procedural Terrain Modeling Framework. In Poster Proceedings of the 14th Eurographics Symposium on Virtual Environments EGVE08, Eindhoven, The Netherlands.

[16] Snavely, N., Seitz, S.M., and Szeliski, R., 2008, "Modeling the world from internet photo collections", *International Journal of Computer Vision IJCV.* vol. 80, 189-210.

[17] Triggs, B., McLauchlan, P., Hartley, R. and Fitzgibbon, A., 1999, "Bundle adjustment - A Modern Synthesis", *Vision Algorithms: Theory and Practice,* 153-177.