Automated Image-Based Abstraction of Aerial Images

Amir Semmo, Jan Eric Kyprianidis, Jürgen Döllner

Hasso-Plattner-Institut, University of Potsdam,
doellner@hpi.uni-potsdam.de

Abstract. Aerial images represent a fundamental type of geodata with a broad range of applications in GIS and geovisualization. The perception and cognitive processing of aerial images by the human, however, still is faced with the specific limitations of photorealistic depictions such as low contrast areas, unsharp object borders as well as visual noise.
In this paper we present a novel technique to automatically abstract aerial images that enhances visual clarity and generalizes the contents of aerial images to improve their perception and recognition. The technique applies non-photorealistic image processing by smoothing local image regions with low contrast and emphasizing edges in image regions with high contrast. To handle the abstraction of large images, we introduce an image tiling procedure that is optimized for post-processing images on GPUs and avoids visible artifacts across junctions. This is technically achieved by filtering additional connection tiles that overlap the main tiles of the input image. The technique also allows the generation of different levels of abstraction for aerial images by computing a mipmap pyramid, where each of the mipmap levels is filtered with adapted abstraction parameters. These mipmaps can then be used to perform level-of-detail rendering of abstracted aerial images.
Finally, the paper contributes a study to aerial image abstraction by analyzing the results of the abstraction process on distinctive visible elements in common aerial image types. In particular, we have identified a high abstraction potential in landscape images and a higher benefit from edge enhancement in urban environments.

1 Introduction

Since its first advent in 1858, when the French photographer and balloonist Gaspard-Félix Tournachon captured aerial images over Paris, aerial photography has evolved to a fundamental tool for geodata capturing, processing, and visualization.

With today's usage of aerial photography in photorealistic visualization, aerial images have become a viable medium in applications such as in cartography, photogrammetry, construction, planning, and marketing. They are also used as popular source of geoinformation in today's geographic information systems (GIS) and by the general public, in particular since the wide spread application of aerial photography in Internet mapping services such as GoogleEarth or Microsoft Bing Maps.
Furthermore, the technical advances in the field of oblique imagery (Pictometry, 2009) have opened new avenues for efficient 3D geoinformation acquisition.

Simultaneously, with today's technical advances in remote sensing for aerial photographs (e.g., aerial images with a resolution of 10 cm per pixel Blom Pictometry, 2009), aerial images have been established as a fundamental data source in 3D geovisualization for constructing realistic, detailed models of reality.

1.1 Non-Photorealistic Spatial Communication

Aerial images show the richness of visual details and the natural appearance of our environment. Insofar, they serve as important geodata source for photorealistic visualization. Non-photorealistic visualization, however, can be a better choice for spatial perception, analysis, understanding and knowledge discovery (Meng, 2002; Nienhaus, 2006) in many areas of application. Those computer-generated presentations basically build on the intention for communicating highly complex contents by emphasizing subtle attributes and features and omitting extraneous information. A typical approach is the imitation of classical depiction techniques in terms of illustrative and aesthetically pleasant visualizations. For example there has been extensive research on visualizing images using computer-generated watercolor effects (Curtis et al., 1997), pencil or pen and ink drawings (Gooch and Gooch, 2001).

Fig. 1. Examples of abstracted aerial images created using our framework

Another classical area of research is the stylization and abstraction of photographs using edge-preserving smoothing and enhancement filters (e.g., Kyprianidis and Döllner, 2008; Kyprianidis et al., 2009; Winnemöller et al., 2006) for depicting bold edges and large regions of constant color. Research in visual perception argues that non-photorealistic rendering should take into consideration the specific nature and type of objects displayed, since "meaningful abstraction clearly affected viewers in a way that supports an interpretation of enhanced understanding" (Santella, 2005). Additional evaluations proof that non-photorealistic visualization has the ability to direct a viewers gaze and focus the interest on particular areas. These are not just applicable to art, but also to wider problems of graphical illustration and visualization (Santella and DeCarlo, 2004), e.g. 3D geovirtual environments. Earlier research on non-photorealistic rendering of 3D geovirtual environments involves the non-photorealistic rendering of expressive representations of 3D city models (e.g., Baumann et al., 2005; Döllner et al., 2005; Döllner, 2007; Glander and Döllner, 2009) and how users can leverage from these expressive depictions in graphics design, primary valuable in application areas like city and landscape planning or tourist information systems

1.2 Abstracted Aerial Images for Non-Photorealistic 3D Geovirtual Environments

Research in the fields of non-photorealistic rendering primarily focuses on rendering of 3D models such as terrain models and 3D building models

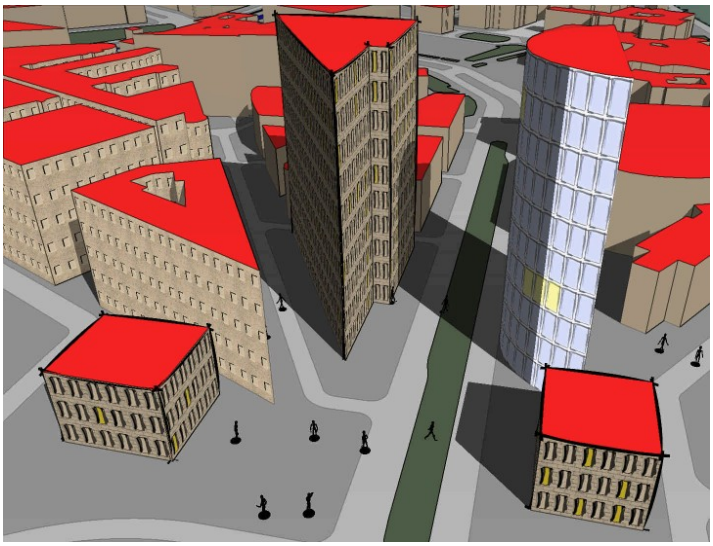

Fig. 2. Illustrative visualization of a 3D city model with stylized edges, facade textures and regular shadows as described by Döllner et al. (2005)

(see Figure 2), but has not developed specific solutions for a proper stylistic representation of the terrain with projected aerial images.
In this work we present a technique for automatically abstracting aerial images in a non-photorealistic way (see Figure 1); the results supply a concise presentation of the terrain in non-photorealistic renderings of 3D geovirtual environments.

Additionally, common image abstraction techniques focus on image filtering that can be processed as a whole by the graphics processing unit (GPU). Our framework, in contrast, contributes an approach to filter massive image data on the GPU in order to allow for post-processing of very large aerial images in a seamless way. Technically we make use of image tiling, which we arrange as main and connection tiles for preventing visible artifacts at junctions. With this procedure we facilitate pre-processing of stylistic image representations such that rendering is feasible on low-cost computing devices or devices with limited computing power, e.g., smart phones or PDAs. To show the potential of abstraction, we analyze the results on distinct common elements that regularly occur in aerial images, e.g., vegetation, buildings, and transportation objects. Our analysis focuses on the abstraction potential in detailed structures and the benefits of edge enhancement that occurs in image discontinuities.

For filtering, we use as initial position primarily those aerial photographs that correspond to two classes: Oblique photos and orthophotos.

Oblique photos are images that are captured at an angle of about 40 degrees from four principle directions. Orthophotos are most valuable to cartography and city planning and are the primary source for GIS to create maps because of their nature, i.e., having a uniform scale and providing undistorted geometry.

2 Related Work

The presented image filtering technique is based on work by Kyprianidis and Döllner (2008), who describe automatic non-photorealistic image processing for creating stylistic illustrations from color images. The technique extends the approach of Winnemöller et al. (2006) to use iterated bilateral filtering for abstraction and difference-of-Gaussians (DoG) for edge extraction by adapting these filters to the local orientation of the input image. A detailed survey on bilateral filtering can be found in (Paris et al., 2007). The Difference-of-Gaussians (DoG) filter is an approximation of the Laplacian-of-Gaussian (Marr and Hildreth, 1980).

A standard approach for dealing with the rendering of large textures is texture tiling. Tanner et al. (1998) introduce the technique of texture clipmaps, a mechanism that virtualizes mipmap textures and is able to manage high-resolution terrain textures by clipping the mipmaps into regions that are updated according to the viewer's movements. A more general approach is introduced by Döllner et al. (2000), who present a texture hierarchy based on a multiresolution model that incorporates for each texture layer an image pyramid and texture tree. They utilize multi-pass rendering and multi-texturing to get real-time performance and texture layer combinations.

3 System Overview

From a technical perspective, there are two crucial aspects for implementing post-processing techniques for aerial images: First the massiveness of the data and second limited hardware resources for processing these data. For example, if the city of London (2.90 km²) would be captured at a resolution of 10cm per pixel (24-bit), we would have to deal with an uncompressed dataset of about 20GB. And as the image post-processing is usually carried-out by the GPU in order to take advantage from parallel

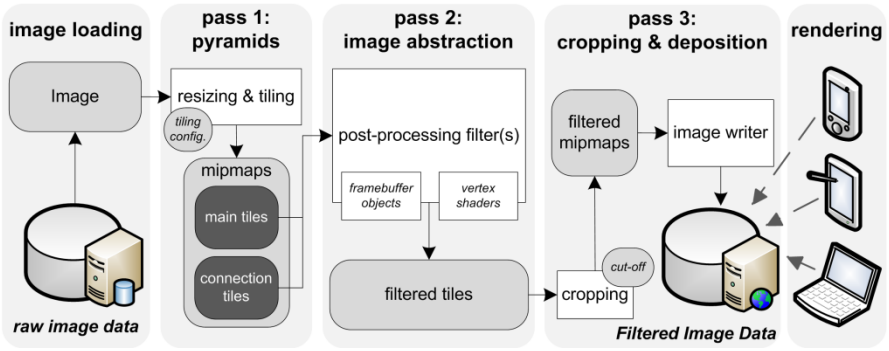

Fig. 3. Overview of our abstraction pipeline. Raw image data (e.g. GeoTIFFs) is loaded and pre-processed by image resizing and tiling. A filtering is conducted on each of the tiles, which are cropped and written back into memory. Finally, the filtered mipmaps are stored onto disk, from where they can be requested for rendering onto different devices

processing, the GPU itself (respectively the texture memory) turns out to be the main bottleneck. Typically GPUs of today's generation can address and process images with 8,192 x 8,192 pixels at the same time.

In our approach, the technique is designed in such a way that these GPU limitations can be by-passed by means of image tiling. Figure 3 illustrates the abstraction pipeline, which we also use for the analysis given in Section 4. The system starts to incrementally load a given aerial image into main memory. Next, the image is tiled in evenly sized main parts such that the GPU can process them as a whole (see Section 3.1). Each of these tiles (or the whole image) are resized down to 1 x 1 pixels, capturing the halve side length images of the upper levels. Those layers correspond to the mipmaps of a usual mipmapping procedure (Williams, 1983), with the difference that we are able to achieve a level of abstraction as each layer will be filtered apart from the other layers (see Section 3.2). After this, the images tiles are loaded into texture memory and filtered by the abstraction stages as described by Kyprianidis and Döllner (2008). For the abstraction process we use off-screen buffers using GLSL shaders with OpenGL. Additional (connecting) tiles that overlap the main tiles are taken into account. This is done to avoid visible artifacts at the junctions of the main tiles during the rendering process (see Section 3.1). This includes the cropping of each of the tiles in the third pass of the pipeline (Figure 3).

The storage of the filtered mipmaps can be handled differently. We compress the filtered mipmaps by the S3 Texture Compression algorithm (S3 Corporation) and store them as DDS (Direct Draw Surface) file such

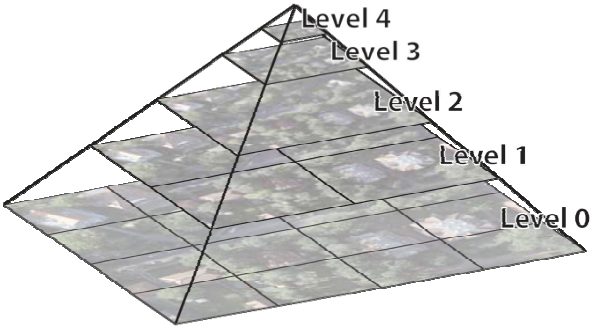

Fig. 4. Overview of an image pyramid consisting of a base image (level 0) and a series of successively smaller sub-images, each at half the resolution of the previous image

that the images/tiles can be rapidly loaded into texture memory and rendered (see Section 3.3). An alternative method could base the storage on a streaming procedure that makes use of the TIFF file standard, which supports storage of mipmaps.

The abstraction pipeline can be handled by a system that provides the abstracted images pre-tiled and filtered on each of the mipmap levels. Computing devices that do not have the power to compute these images in an adequate time span, or do not provide the required hardware, can request these images and render them directly on-screen.

3.1 Image Resizing & Tiling

After loading, the system scales down the aerial image on each mipmap level and tiles those layers in smaller parts. Most web services providing aerial images already have them split in evenly sized tiles; mapping services for example prefer a tile dimension of 256 or 512 pixels. In our approach, we do not assume to have a pre-tiled image and partition the scaled mipmaps in tiles with a size of 1,024 x 1,024 or (if possible) 2,048 x 2,048 pixels. Image tiling and resizing are processes that are not performed in a distinctive order, but usually images are tiled first and resized afterwards. In any case, this results in an image pyramid that looks like the one in Figure 4.

The plain abstraction of image tiles by filtering in local image domains, however, will cause visible artifacts at the junctions as soon as the scene is being reconstructed and rendered (Figure 5). By taking additional tiles into

Fig. 5. A visible junction between two tiles in case a plain filtering is conducted. Edges are apparently interrupted and colors abruptly change

account that overlap the junctions, we are able to deal with this problem. Conceptually, each one of these connection tiles overlap two adjacent main tiles in each direction by a cut-off value that is adapted to the domain range of the bilateral filter, which is defined by its standard deviation σ_d. For calculating the lowest cut-off, we will also need to consider the number of iterations of the bilateral filter n_a:

$$\lceil 2 \cdot \sigma_d \rceil \cdot n_a$$

The connection tiles can occur in each main direction of the area as well as in the center of 4 adjacent main tiles, where we take an additional smaller squared tile into account. The dimensions of these tiles can be seen in Figure 6.

3.2 Image Abstraction

For image abstraction, we load each of the tiles into texture memory and run the following post-processing stages on input and temporary generated data. For further details on each of the stages and optimizations we refer to Kyprianidis and Döllner (2008):

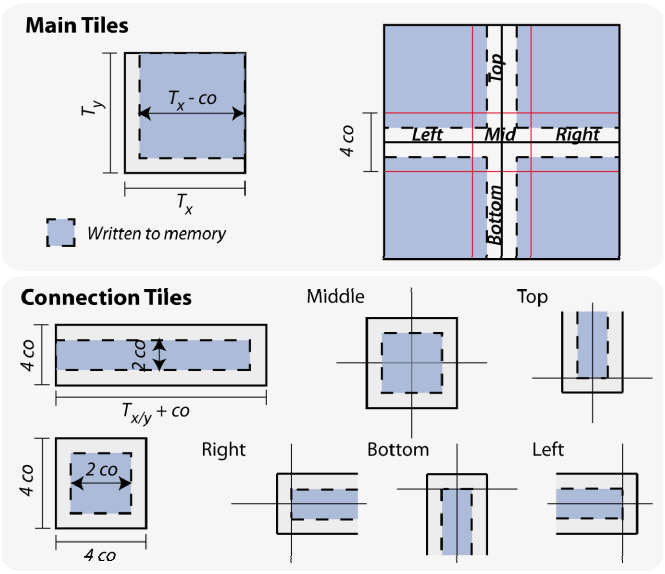

Fig. 6. Overview of the cropping of main tiles and connection tiles for encountering the problem of visible artifacts at junctions. $(T_x \times T_y)$ denotes the size of image tiles

1. Local Orientation Estimation: The estimation of the local orientation of an image allows us to obtain information about the dominant orientation of a region. This information is used for optimizing the bilateral and difference-of-Gaussians filter. It is mathematically retrieved from the structure tensor and its eigenvectors, which encode information about a region's structure.
2. Bilateral Filter: We use the bilateral filter as the main construct for our image abstraction. It replaces the pixel's value by a weighted average of its neighbors in both, space and intensity, and has the characteristic to smooth an image while preserving edges.
3. Edge Detection: A flow-guided anisotropic difference-of-Gaussians kernel, whose shape is defined by the local orientation, is used for edge detection. As first pass, we apply a one-dimensional DoG filter in gradient direction followed by a second pass that applies smoothing along the flow curves of the vector field induced by the smoothed structure tensor.
4. Color Quantization: As last step, we quantize the color range of the abstracted image by using the smoothed step function from (Winnemöller et al., 2006). The final output of this stage is then combined with the product of the edge detection.

Fig. 7. Level of abstraction by means of trilinear filtering. The viewpoint distance decreases linearly from left to right. The greater the viewpoint distance, the fewer details are depicted at the roof of the building and the tree

The whole process is conducted on each of the mipmaps of a tile. With this procedure we are able to establish a level of abstraction by using the trilinear texture mapping capabilities of the GPU. A result can be seen in Figure 7, where subtle details like roof tiles are completely filtered out from 25% of the original image size, outlines shadows are increasingly coarsened and complex objects shapes like bushes or treetops are combined to simpler structures. This is due to the nature of the bilateral filter, having much less pixels of a distinct structure in a resized image left for weighting the color values into the local neighborhood.

3.3 Image Compression & Storage

For image compression and storage we make use of the nVidia Texture Tools library (see NVIDIA Texture Tools 2) for storing the filtered mipmaps as DDS (Direct Draw Surface) files and stitching them to a single DDS file. The DDS scheme is an optimized data format for storing mipmaps and DXT compressed textures. This allows us to render the image tiles quickly with moderate memory consumption.

4 Analysis of Abstracted Aerial Images

To show the feasibility of automated aerial image abstraction, we analyze the results of our approach by distinguishing between the visible objects in aerial images. This is done because the process of abstraction and edge enhancement scales differently among the distinct visible objects, whose characteristic illustration mainly depict similar structures across different images.

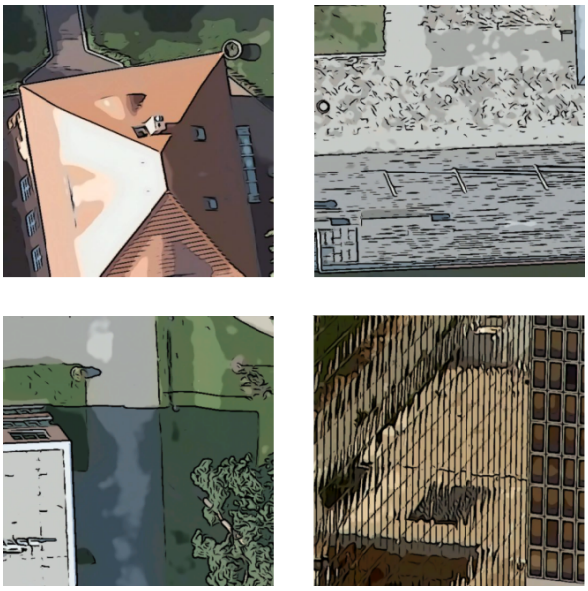

Fig. 8. The four categories of discontinuities that can occur in aerial images and affect the edge detection

On the one hand, this analysis focuses on the abstraction potential in fine granular structures for filtering extraneous information. On the other hand, our analysis identifies benefits from edge enhancements that are derived from four different discontinuities that regularly occur in aerial images: Discontinuities in depth, surface orientation, illumination and reflectance. Those discontinuities primarily highlight well-defined boundaries of objects and allow, due to contrast differences, a detection of edges and enhancement for a non-photorealistic representation (see Figure 8).

Our analysis is based on the thematic model of the CityGML (OpenGIS City Geography Markup Language) specification and the categorization of Döllner et al. (2005). For an overview, we refer to Figure 9 and Figure 10.

4.1 Relief & Terrain Objects

Relief and terrain represent the essential element of aerial images, especially in landscapes photographs. We count everything as plain terrain that reveals the rock of the earth and does not feature unnatural structures, e.g., sand, split, or gravel. Dependent on the granularity of the terrain, the abstraction may unfold a high potential for filtering extraneous information. For example, sandy soils can mainly be depicted as single-colored planes,

especially when no edge detection is conducted and the color quantization is limited to few colors. On the other hand, structures with a high granularity like mountainous areas benefit from their three-dimensional structure as light spots get caught and accordingly strengthen the color differences of the terrain's surface.

Additionally, lighting plays the main role for highlighting variations in the slope of the surface and correspondingly supports the process of edge enhancement for depicting the characteristic shapes like trenches and crests. This process can be traced back on discontinuities in the surface orientation, for example when different rock faces meet or at the boundary of snow lines in mountainous areas.

4.2 Vegetation Objects

Vegetation objects always have been a challenge for modeling in 3D geovirtual environments due to their complex structure and fuzziness (see Foody, 1996; Muhar, 1999). Basically we distinguish between the range and 3-dimensional structure (physiognomy) of plants, e.g., trees and bushes as rangy plants and grass as open and plane vegetation. With this categorization we achieve considerable variations in the depiction of abstraction. Lawn for example can induce the highest potential of abstraction just as in the case of fine granular terrain structures. On the other hand, treetops are structures that can highly leverage from the level of abstraction concept. In this case, the edge enhancement turns out to be the main depiction technique, since sub-branches of the tree and their foliage are mainly depicted as partial clouds that primarily differ from their surrounding by their difference in height - an effect that is reinforced by the actual direction of lighting. As the viewing distance increases, the differences in the structures decrease and sub-leafages more and more merge.

4.3 Water Surface Objects

Water surfaces can be described as boundaries that occur between water and air. We count water surfaces as the third element with high abstraction potential, because generally they can be depicted as single-color planes without any edge enhancement. Nevertheless, due to high specular characteristics of water surfaces, they tend to feature areas with high reflectivity, whose abrupt color differences to the normal surface color might influence an edge enhancement process.

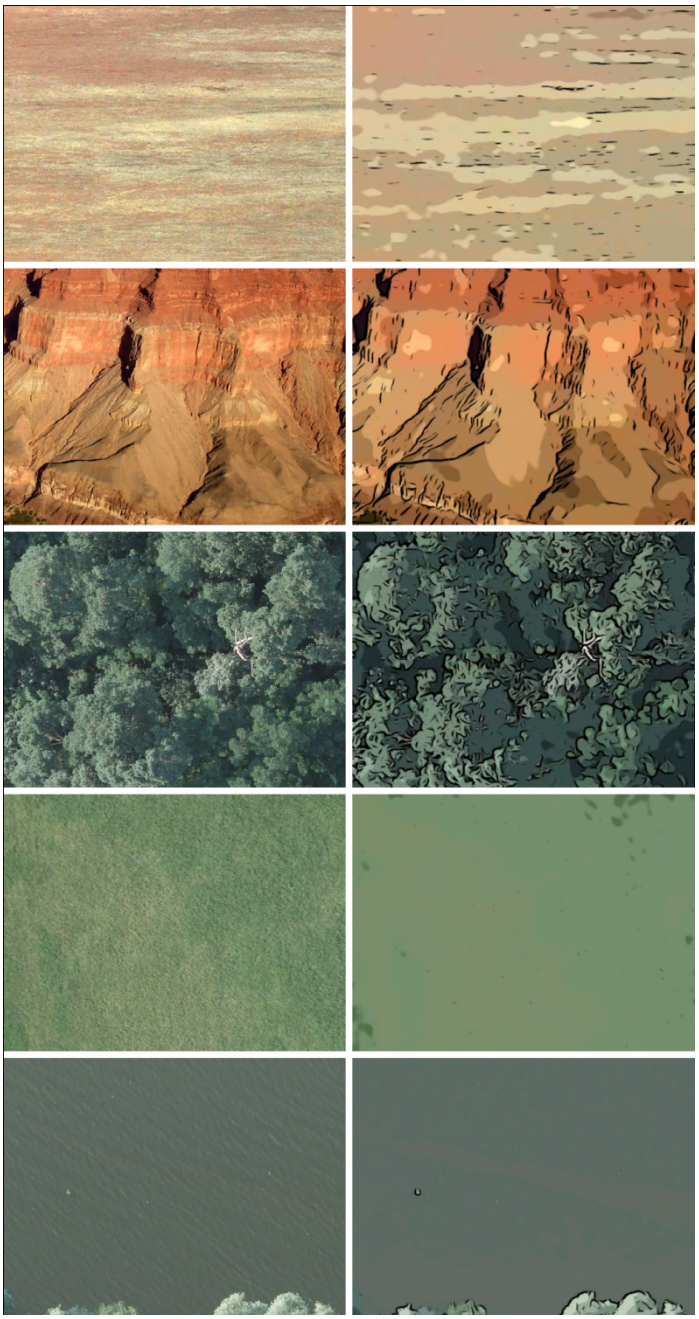

Fig. 9. Common object types that occur in aerial images. Top: Relief and terrain. Middle: Vegetation objects. Bottom: Water surfaces

Fig. 10. Common object types that occur in aerial images. Top: City furniture. Middle: Building and sites. Bottom: Transportation and population objects

4.4 Building & Site Objects

The major illustrated elements of a building are the roofs, facades and installations, whose richness of detail highly depends on the viewing distance and lighting condition. Regarding the roofs, usually each one of the sides is depicted by its own major color tone. The edge enhancement highlights the roof-edges and illustrates a suggestion of the roof tiles due to discontinuities in the surface orientation. In any case the exterior shell and distinctive shape of the roof is reflected due to its difference in elevation from the surrounding (depth discontinuities, see Figure 8). Regarding the facades we usually have the same effect as for the roof: Plastering or fine granular textures are depicted as single colored areas and objects like windows are enhanced as squares.

4.5 Transportation Objects

Transportation objects refer to all objects that are dedicated for the movement, traffic and transport, e.g., roads, rails or pavements. Usually roads and pavements can be abstracted with wide filtering kernels and a high color quantization process. The difficulty occurs when those objects are roughly structured, for example just like in the case of cobbled streets. Here the abstraction needs to operate on smaller local areas and the edge enhancement ideally takes a higher abstracted buffered image as input in order to operate more abrasive but with the characteristic to not enhance noise that occur on the surface, especially in the case of rapid changes that occur in the surface orientation (Figure 8).

4.6 City Furniture Objects

City furniture denotes an inhomogeneous group of objects that are statically placed with special function related to traffic, transportation, decoration, or advertisement. Examples are street lights, traffic signs and lights, benches, garbage cans and billboards for advertisement. Due to their architectural structure, they similarly behave like the abstraction of buildings and sites.

4.7 Population Objects

We count for example cars or people as population objects. These elements are usually the least important details in aerial images that should be

kept in an abstraction process. Unfortunately due to its size, objects like cars will be obtrusively linked with the surrounding. Recent work studied the recognition of those objects (Leberl et al., 2007; Ram et al., 2001). Results in this area might facilitate a pre-processing that is able to remove those objects in certain conditions.

5 Performance

We used the following default values for the filtering process, whose configuration is feasible for pleasant results of any type of aerial image. The flow field of the local orientation estimation is smoothed by a standard deviation of 2.0 in order to remove discontinuities. For the bilateral filter the first iteration is used for edge extraction and the fourth iteration as input for the color quantization. The size of the kernel has a standard deviation of 3.0, whereas the amount of smoothing the bilateral filter applies at edges is 4.25%. For the edge detection we conducted one iteration. To balance bandwidth and sensitivity of the difference-of-Gaussians filter, standard deviations 1.0 and 1.6 are used, as recommended by Marr and Hildreth (1980). We used a standard deviation of 3.0 for the smoothing filter applied along the streamlines. Our color quantization outputs at most 8 distinct colors for a local region. We additionally conducted a usual Gaussian-smoothing on the final output with a kernel size of 3x3 pixels. Figure 11 depicts an output that is based on these values.

The percentage of connection tiles in relation to the input size of an image can be calculated as follows. Let $I_x \times I_y$ be the dimension of the input image and let $T_x \times T_y$ be the dimension of main tiles. Then the number of main tiles in x- and y-direction are given by

$$n_{Tx} = \frac{I_x}{T_x} - 1 \quad \text{and} \quad n_{Ty} = \frac{I_y}{T_y} - 1 \ .$$

The total size in pixels for the connection tiles is:

$$p_t(I_x, I_y) = n_{Tx} \cdot n_{Ty} \cdot 4 \cdot co^2 + \left(n_{Tx} + n_{Ty} + 2 \cdot n_{Tx} \cdot n_{Ty}\right)\left(4 \cdot co \cdot (I_x + I_y + co)\right)$$

The number of these connection tiles increase nonlinear with the dimension of the input image $I_x \times I_y$. Nevertheless, in case of a tiling size of

2,048 x 2,048 pixels and cut-off of $co = 24$ pixels, the percentage levels off at a limit of approximately 18.9%. GPUs that would be able to post-process image tiles of 8,192 x 8,192 pixels would end up with an off-cut of 4.7% (2.3% with $co = 12$).

An empirical analysis of the run-time emphasizes the decreased overhead if larger main tiles are conducted. As filtering settings we used the default configurations described previously on a test system with a Core2 Duo E8400 3.0GHz and a nVidia GeForce 9600 GT. Images with a size of 8,192 x 8,192 pixels took in mean 21.1 seconds to process for a tiling of 1,024 x 1,024 pixels (16.5s with $co = 12$), whereas the same image took 17.8 seconds to process for image tiles of 2,048 x 2,048 pixels (15.4s with $co = 12$).

6 Conclusions & Future Work

In this paper, we have presented an approach for automated, image-based abstraction that can handle massive, tiled aerial photography and that takes advantage of a GPU for efficiently performing image filtering.
We have shown that the abstraction of aerial images result in visually pleasant depictions. By analyzing seven different object types and identifying the impact of a filtering configuration on each of the elements' structure, we have shown how different content types of aerial photography can be graphically abstracted. In addition, we have shown how a concept of levels of abstraction can be established. Overall, our work contributes to the feasibility of non-photorealistic, automated aerial image abstraction as a pre-processing stage, required by applications such as for mobile devices with a low computing power and limited display size.

Primary application areas include non-photorealistic representations of 3D city models, which can take advantage from proper terrain stylization. As future work, we would like to explore the acceptance of abstracted aerial images as orientation guidance in mapping services like Google Maps or Microsoft Bing Maps, especially in areas that are little-known to a user. A second topic could deal with the quality of abstraction by conducting a dynamic filtering, which could be based on the actual mapped content. We could do so by distinguishing between primarily landscape or aerial images, or tie it to the wide topic of automatic object recognition (e.g. Mayer, 1999; Zhao et al., 2008; Idbraim et al., 2008) for combining it with our object categorization from Section 4. In this context it would also be worthwhile to investigate other abstraction filters that preserve edges, e.g., the anisotropic Kuwahara filter by Kyprianidis et al. (2009).

Fig. 11. An example created by our abstraction framework. Image tiles: 1,024 x 1,024 pixels

Figure 8 (bottom-right) original photo by Clearly Ambiguous - http://www.flickr.com/photos/clearlyambiguous
Figure 9 (top-left) original photo by mistress_f - http://www.flickr.com/photos/mistressf
Figure 9 (second from top-left) original photo by <<graham>> - http://www.flickr.com/photos/schnappi
Figure 10 (top-left) original photo by Daquella manera - http://www.flickr.com/photos/daquellamanera
Figure 11 (top-left) original photo by IRRI Images - http://www.flickr.com/photos/ricephotos

References

Bigün, J. and Granlund, G. H. (1987) Optimal orientation detection of linear symmetry, Proceedings of the IEEE First International Conference on Computer Vision, London, Great Britain, pp. 433-438.

Blom Pictometry (2009) http://www.blompictometry.com, Last date accessed 11.2009.

Buchholz, H., Döllner, J., Nienhaus, M. and Kirsch, F. (2005) Real-Time Non-Photorealistic Rendering of 3D City Models, 1st International Workshop on Next Generation 3D City Models, EuroSDR.

Curtis, C. J., Anderson, S. E., Seims J. E., Fleischer, K. W. and Salesin, D. H. (1997) Computer-generated watercolor, SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques, New York, pp. 421-430.

Döllner, J. (2007) In: Non-Photorealistic 3D Geovisualization, Multimedia Cartography, Springer, pp. 229-240.

Döllner, J., Baumann, K., Hinrichs, K. (2000) Texturing techniques for terrain visualization. In: Proceedings of IEEE Visualization, pp. 227-234.

Döllner, J., Buchholz, H., Nienhaus, M. and Kirsch, F. (2005) Illustrative Visualization of 3D City Models, In: Visualization and Data Analysis, Proceedings of the SPIE, International Society for Optical Engine (SPIE), pp. 42-51.

Foody, G. M. (1996) Fuzzy modelling of vegetation from remotely sensed imagery, In: Ecological Modelling, vol. 5669, pp. 3-12.

Glander, T. and Döllner, J. (2009) Abstract representations for interactive visualization of virtual 3D city models, In: Computers, Environment and Urban Systems

Gooch, B. and Gooch, A. A. (2001) Non-Photorealistic Rendering, AK Peters Ltd.

Idbraim, S., Mammass, D., Aboutajdine and Ducrot, D. (2008), An automatic system for urban road extraction from satellite and aerial images, In: WSEAS Trans., Sig. Proceedings, vol. 4, no. 10, pp. 563-572.

Open Geospatial Consortium (O.G.C.) (2008) OpenGIS City Geography Markup Language (CityGML) Implementation Specification, http://www.opengeospatial.org/standards/citygml, http://www.citygml.org, Last date accessed 11.2009.

Kuwahara, M., Hachimura, K, Eiho, S. and Kinoshita, M. (1976) Digital processing of biomedical images, Plenum Press.

Kyprianidis, J. E. and Döllner, J. (2008) Image Abstraction by Structure Adaptive Filtering, In: EG UK Theory and Practice of Computer Graphics, Eurographics Association, pp. 51-58.

Kyprianidis, J. E., Kang, H. and Döllner, J. (2009) Image and Video Abstraction by Anisotropic Kuwahara Filtering, Computer Graphics Forum, vol. 28, no. 7.

Leberl, F., Bischof, H., Grabner, H. and Kluckner, S. (2007) Recognizing cars in aerial imagery to improve orthophotos, In: GIS '07: Proceedings of the 15th annual ACM international symposium on Advances in geographic information systems, New York, pp. 1-9.

Marr, D. and Hildreth, E. (1980) Theory of edge detection, RoyalP, vol. B-207, pp. 187-217.

Mayer, H. (1999) Automatic object extraction from aerial imagery - A survey focusing on buildings, Computer vision and image understanding, vol. 74, no. 2, pp. 138-149.

Meng, L. (2002) How can 3D Geovisualization Please Users Eyes Better?, Geoinformatics Magazine for Geo-IT Professionals, Emmeloord, The Netherlands, vol. 5, pp. 34-35

Muhar, A. (1999) Three-dimensional modeling and visualization of vegetation for landscape simulation, Institute for Landscape Architecture and Landscape Management, Ascona, Switzerland, Technical report.

Nienhaus, M. (2006) Real-Time Non-Photorealistic Rendering Techniques for Illustrating 3D Scenes and their Dynamics, Ph.D. dissertation, HPI, Universität Potsdam, Germany, 2006.

NVIDIA Texture Tools 2 - GPU-accelerated Texture Tools with support for DirectX 10 texture formats (2009) http://code.google.com/p/nvidia-texture-tools, Last date accessed 11.2009.

Paris, S., Kornprobst, P., Tumblin, J. and Durand, F. (2007) A gentle introduction to bilateral filtering and its applications, In: SIGGRAPH '07: ACM SIGGPRAPH 2007 courses, New York, p. 1.

Pictometry - The Aerial Oblique Photography Company (2009) http://www.pictometry.com, Last date accessed 11.2009.

Ram, T. Z., Zhao, T. and Nevatia, R. (2001) Car detection in low resolution aerial images, In: Image and Vision Computing, pp. 710-717.

Santella, A. (2005) The Art of Seeing: Visual Perception in Design and Evaluation of Non-Photorealistic Rendering, Ph.D. dissertation, Rutgers University, New Brunswick, New Jersey, USA.

Santella, A. and DeCarlo, D. (2004) Visual interest and NPR: An evaluation and manifesto, In: NPAR '04: Proceedings of the 3rd international symposium on Non-Photorealistic animation and rendering, New York, USA, pp. 71-150.

Tanner, C. C., Migdal C. J. and Jones, M. T. (1998) The clipmap: A virtual mipmap, In: SIGGPRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques, New York, USA, pp. 151-158.

Tomasi, C. and Manduchi, R. (1998) Bilateral filtering for gray and color images, IEEE International Conference on Computer Vision (ICCV), p. 839.

Williams, L. (1983) Pyramidal parametrics, In: SIGGRAPH '83: Proceedings of the 10th annual conference on Computer graphics and interactive techniques, New York, pp. 1-11.

Winnemöller, H., Olsen, S. C. and Gooch, B. (2006) Real-time video abstraction, In: ACM Trans. Graph., vol. 25, no. 3, pp. 1221-1226.

Zhao, F., Zhang, H., Li, Z. and Pang, Y. (2008) The extraction of individual tree-crown in aerial digital camera imagery, In: FSKD (3), pp. 183-188.