Automatic Extraction of Man-Made Objects from Aerial and Space Images (II)

Advancements in digital sensor technology, digital image analysis techniques, as well as computer software and hardware have brought together the fields of computer vision and photogrammetry, which are now converging towards sharing, to a great extent, objectives and algorithms. The potential for mutual benefits by the close collaboration and interaction of these two disciplines is great, as photogrammetric know-how can be aided by the most recent image analysis developments in computer vision, while modern quantitative photogrammetric approaches can support computer vision activities.
Devising methodologies for automating the extraction of man-made objects (e.g. buildings, roads) from digital aerial or satellite imagery is an application where this cooperation and mutual support is already reaping benefits. The valuable spatial information collected using these interdisciplinary techniques is of improved qualitative and quantitative accuracy. This book offers a comprehensive selection of high-quality and in-depth contributions from worldwide leading research institutions, treating theoretical as well as implementational issues, and representing the state-of-the-art on this subject among the photogrammetric and computer vision communities. These contributions were presented at a Monte Verità Workshop organised in May 1997, following a first Workshop on the same topics in April 1995.

9 783764 357887

ISBN 3-7643-5788-6

Monte Verità
Centro Stefano Franscini Ascona
ETH Zürich

Automatic Extraction of Man-Made Objects from Aerial and Space Images (II)

Edited by
A. Gruen
E.P. Baltsavias
O. Henricsson

Birkhäuse

Fig. 4 A perspective view showing the tower view.

Fig. 5 A perspective view showing taxiway signs.

Automatic Extraction of Man-Made Objects from Aerial and Space Images (II) Monte Verità, © 1997 Birkhäuser Verlag Basel

Managing Large 3D Urban Database Contents supporting Phototexture and Levels of Detail

Michael Gruber°, Michael Kofler° and Franz Leberl*

°Institute for Computer Graphics and Vision
Technical University Graz
Münzgrabenstraße 11, A-8010 Graz, Austria
Ph: ++43 (316) 873-5011, Fax: ++43 (316) 873-5050
e-mail: {gruber, kofler}@icg.tu-graz.ac.at

*Austrian Research Center Seibersdorf
A-2444 Seibersdorf
Ph: ++43 (2254) 780-2001, Fax: ++43 (2254) 780-2010
e-mail: leber@arcs.ac.at

Abstract

We present a method to handle large database contents for visualization of urban scenes. These data consist of geometry (vector data) and phototexture from images (raster data) which have different size and different characteristics. A high emphasis is given to the possibility of remote use (client/server system) and therefore of special concepts to manage, retrieve, transfer and visualize data from local and remote sites.

Data are organized and retrieved exploiting the concepts of a three dimensional R-tree, containig different levels of detail and texture from images stored under different resolutions. We show our results concerning two test sites, the city of Vienna and the province of Styria.

1. Introduction

Observing the evolution of photorealistic 3D digital models of real world objects, we recognize a tremendous development of computing hardware. Workstations of a capability of graphics supercomputers from three years ago are now available as desktop machines. Texture from images is stored and prepared for rendering at specific hardware components; software is available to manage and organize these data in a very effective way [1, 4, 6, 7].

The development of high performance network facilities is very promising as well. High speed transfer of large quantities of data will be possible for a lot of users in the near future. Even if this technology is driven and influenced by the entertainment and information industry, the results will be available for other applications.

We have focused on organizing and distributing large database contents like those of a 3D GIS of an entire city or province, containing geometry and texture of terrain, vegetation and man made objects. Practical experiences have been made by creating and manipulating several test data sets, which represent different levels of detail and quality and allow to calculate the total amount of data needed for an entire city like Vienna.

The data base is being prepared not only to create renderings of the scene in an off-line scenario but also to permit interactive walk-trough and fly-over. Therefore, we have to make sure that all necessary data has been selected, transferred and stored on site to be ready for visualization.

2. 3D Digital City Models

Since 3D digital models of urban areas have been created, maintained and used we have observed different requirements in size, level of detail, quality and resolution of texture [3]. In this contribution we try to estimate the total amount of data which is necessary for modeling an entire city like Vienna. We compare four test sites, which have been created at the Institute for Computer Graphics and Vision for different purposes and are prepared for online visualization and interaction (cf. Fig. 1). The source data are aerial images, images from vertical faces (taken from street level) and geometry, which has been reconstructed by traditional methods [2]. Table 1 summarizes the amount of data used for the test sites and estimates how much data are necessary to build a 3D model of the entire city of Vienna at the same quality. Vienna spreads over an area of 400 km^2 and contains about 220000 buildings.

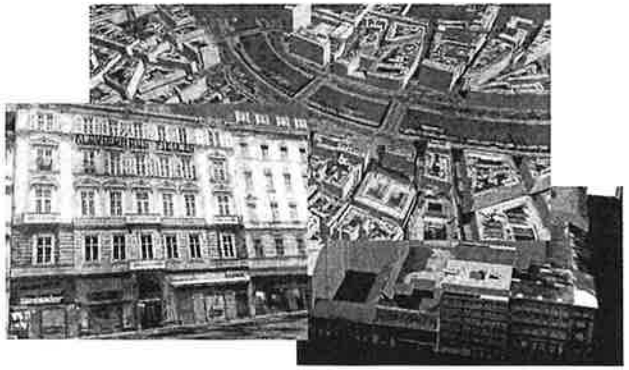


Fig. 1: Digital models of cityscapes presenting different levels of detail

Test Site	Quality (application)	Level of Detail	Resolution of phototexture	Amount of Data (per test site)	Grand Total (Area of Vienna)
Graz	high quality, VR applications	high LoD, bay windows, chimneys etc	4 cm for facades 10 cm for others	90 MByte Texture	500 GByte Tex. 250 MByte Geom.
Vienna 1	3D GIS	medium LoD, building box, roofs	8 cm for facades 20 cm for others	12 MByte Texture	100 GByte Tex. 80 MByte Geom.
Vienna 2	medium scale	low LoD, spare building box	only from aerial images, ~50 cm	24 MByte Texture	1600 MByte Tex. 50 MByte Geom
Vienna 3	overview	low LoD, spare building box	----	50 MByte Geometry	50 MByte Geom.

Tab. 1: Test data sets of different quality and amount of data used for rendering

As soon as phototexture is involved, computers are not able to store the data of an entire city in main memory. Even if only a coarse resolution is available like for test site Vienna 2, it is not possible to load the entire model into main memory. The development of data storage and data retrieval concepts is necessary.

3. 3D database design

We argue that RDBMS (*relational database mangement systems*), as they are used for most of the current GIS applications, are not sufficient for digital 3D city models. The complex structure of three dimensional objects, the simultaneous retrieval of vector and raster data and the need for perspective visualization leads to very special requirements. The tabular approach of the RDBMS appears unqualified for most of these tasks. In contrast, OODBMS (*object oriented data base management systems*) allow to define complex hierarchical data models. Tailored for the specific task, we can create and store data structures which are able to deal with different levels of detail, to fulfill special requirements for perspective visualization and to handle texture data [5].

3.1 The R-tree Data Structure

The R-tree data structure (rectangle tree) is a proven concept to handle large amounts of 2D GIS data in a lot of applications. R-trees can easily be expanded to handle three dimensional data: The data are simply structured into hierarchical 3D boxes instead of rectangles. These overlapping subregions of the entire scene are used as a mechanism to retrieve data within the viewing frustum of a perspective viewport and to choose between different levels of detail (LoDs) and an adequate texture resolution.

In one of our test applications, a walk-through system for a 50 MByte data set of the entire city Vienna (20000 blocks of buildings, more than 1000000 vertices), a 3D R-tree has given a tremendous gain in performance (see Fig. 2 to 4, Table 2). The management of phototexture is not yet implemented in this example.

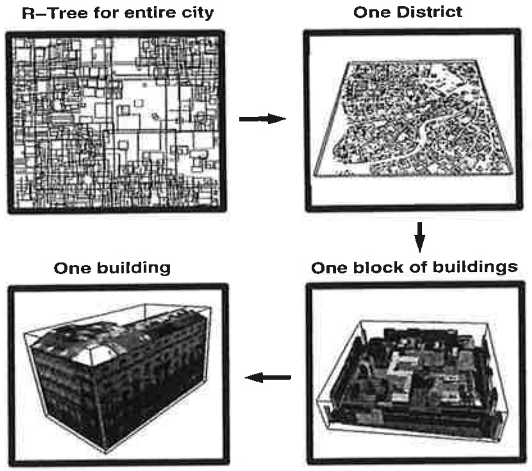


Fig. 2: Example of different data stored in different levels of the R-tree structure

The first visualization was done using Open Inventor. Due to the high memory demands and the internal reorganization of the 3D data it took more than 15 minutes to load the scene using an Indigo2 Silicon Graphics workstation. Visualization then only offered about one frame every 2 seconds. (Open Inventor is the predecessor of VRML. Open Inventor and VRML 1 are compatible.)

To achieve a better performance, the data set then was organized by an R-tree, at the same time using three levels of detail. For each frame, an R-tree query is performed to find out which objects are visible at what distance from the viewpoint. Visible objects near the viewpoint are rendered in full quality, visible object far away from the viewpoint are rendered in less quality, and objects not visible are not even considered for rendering. Instead of Open Inventor the graphics library Open GL is used for rendering.

With this approach we got approximately 10 frames per second using the same hardware! Although the R-tree is built new at startup time, it takes only 20 seconds until the first frame is visible. As the walk-through program dynamically adjusts to the hardware, it is possible to visualize this data set on less expensive machines. (With Open Inventor a visualization was outright impossible with less then 128 MByte of main memory.)

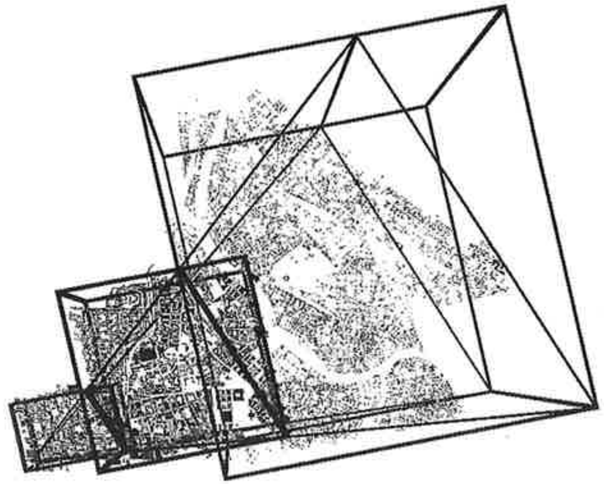


Fig. 3: The viewing frustrum is used to choose between different LoDs

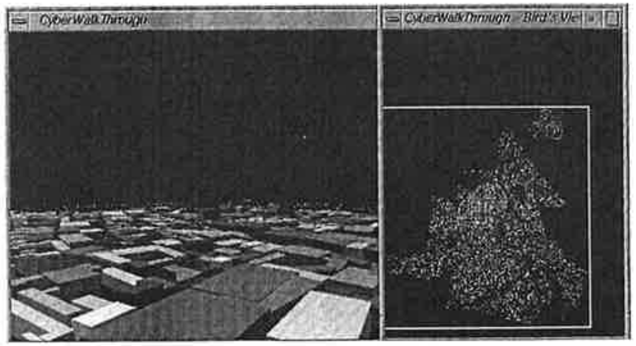


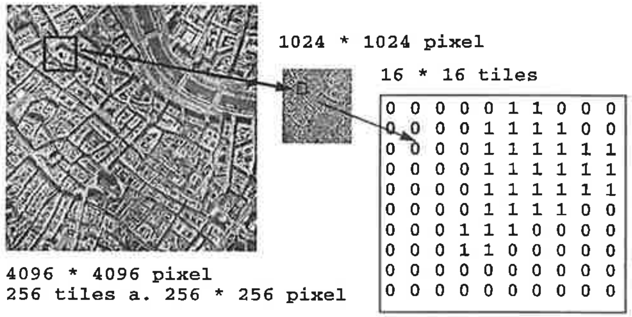
Fig. 4: The walkthrough system in full action

R-Tree Level	Size of 3D boxes	Content of 3D boxes	LoD
Level 0	entire City of Vienna	pointers to Level 1	----
Level 1	100 boxes, approx. 2 *2 km^2	pointers to Level 2, data for LoD3	one pixel (center point)
Level 2	16 boxes, approx. 0.5*0.5 km^2	pointers to Level 3, data for LoD2	minimized bounding box
Level 3	1 building box / building	high quality data (LoD1)	detailed geometry

Tab. 2: Data contents of R-tree levels (example based on the size of the City of Vienna)

3.2 Image Data Management

Digital images are traditionally presented as rectangular arrays of picture elements. In other words and with respect to data management tradition we call them binary large objects (BLOBs). Even if only small images are involved, we have to accept that these data are different from traditional GIS data and need to be dealt in a specific manner. In this contribution we develop a concept to prepare, store and retrieve digital images, which are extracted from vertical aerial images. The test site Vienna 2 is used to demonstrate this concept (see Table 1).



0	0	0	0	0	1	1	0	0	0
0	0	0	0	1	1	1	1	0	0
0	0	0	0	1	1	1	1	1	1
0	0	0	0	1	1	1	1	1	1
0	0	0	0	1	1	1	1	1	1
0	0	0	0	1	1	1	1	0	0
0	0	0	1	1	1	0	0	0	0
0	0	0	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Fig.5: Image data management for test site Vienna 2. Aerial images are subdivided into two parts of 4096 * 4096 pixels. These parts are subdivided into tiles of size 256 * 256 pixels.

We exploit existing three dimensional data of buildings, terrain and aerial images at a scale of 1 : 6000. The aerial images are scanned with a pixel size of 15 μm. The basic idea of our concept is to subdivide the digital image data into tiles and to retrieve these data by computing a minimum number of tiles necessary to cover the area of visibility. A so called neighborhood cache is created to administrate the local texture memory. The digital image data are retrieved from the data base and transferred to this cache on demand. This approach is scaleable, allows to be tuned with respect to the size and power of the envisaged workstation and does not need an orthoimage-like metaproduct. Fig. 5 shows how data of test site Vienna 2 were organized to fit into this concept.

Test site Vienna 2 consists of 3D geometry and six source images covering an area of 6 km^2. We show how these existing data can be used to create a 3D city model of the distinct area. Textures for vertical faces (facades of buildings) are not available. The total amount of texture used in this test site was more than 200 MByte at a ground resolution of 18 cm (source data); the actual model finally incorporated about 12 MByte texture data at a ground resolution of 72 cm.

3.3 Image Data retrieval

It is a key question for the data retrieval task how images are stored and texture data are retrieved for visualization. One approach to handle 3D digital models is the one-face-one-texture concept. Each planar face of the model is directly connected to a distinct texture data file and related texture coordinates are used to compute the parameters for the mapping procedure. Exploiting the functionality of a OODBMS, we can define three dimensional digital objects as a set of geometry, texture and texture coordinates or geometry, texture coordinates and a specific link to the texture data file system. This approach works fine as long as texture data are small (e.g. 512 * 512 pixels or even smaller).

If large texture files are involved, we cannot store and retrieve these data in the same way as small ones. In order to use aerial images as a well known source of texture data and to allow a larger size of these texture we have to fragmentize large files. The fragments are stored in a hierarchy and retrieved on demand. The benefits of this approach are three folded:

- digital aerial images are managed and stored as self-reliant units; updates will be easy, orientation parameters are well known; aerial images are reliable;
- the mapping procedure is based on the set up parameters of the aerial images; no texture coordinates need to be stored in addition to the 3D geometry;
- if each roof had its own texture, the large number of roofs in a city would cause an enormous number of small texture files; the number of aerial images is much smaller and can be maintained more easily.

The procedure of data retrieval is again based on an R-tree query for all objects within the viewing frustum. Each cell of the R-tree contains a set of up to four pointers to the image file system. (The number of pointers depends on the position of the R-tree box relative to the aerial images. Each point of the scene is covered by a maximum of four aerial images.)

The file system contains entire images in a specific tiled structure (full resolution) as well as additional images of the same region in lower resolution (LoD). The R-tree does not contain a mechanism to identify the tiles which are to be used for the texture mapping procedure. In order to find out which tiles are needed for rendering, we build an address array for the tile structure of the image. This array allows us to identify the specific fraction of the entire image which is necessary for visualization. It is updated during the mapping process of low level resolution. Not the entire image but only tiles necessary for rendering are actually transferred to the client machine. Fig. 6 shows the whole procedure of image data retrieval and mapping.

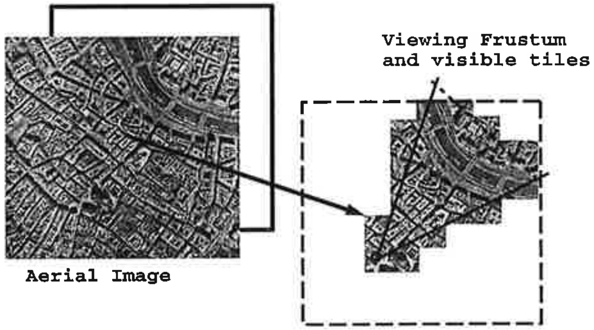


Fig.6: Image data retrieval and address management structure

The benefit of the proposed approach is evident. If we take in account that the neighborhood of a specific viewpoint is covered by at least four aerial images, we would have to transfer 64 MByte of data (assuming 4k * 4k * 8bit digital images). If we transfer only those parts of the images which are needed, the amount of data is reduced to 30 % (according to the example of Fig. 6).

4. A fly-through system for the province of Styria

We have also investigated (but not yet implemented) the possibilities of real time visualization and fly-over system for larger areas such as the province of Styria (17 000 km^2). The digital terrain model contains one point every 50 m (more than 6,000,000 points). Texture data are available in two resolutions: one color pixel every 25 m (40 mil. pixels, Landsat) or one gray shade pixel every 10 m (170 mil. pixels, Spot). With eight bits/pixel, we have to manage at least 170 MByte of texture data (cf. Fig. 7).

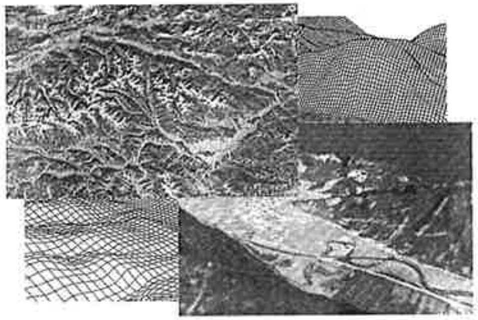


Fig. 7: Source data and 3D textured fraction of the test data set of the province of Styria

Obviously, even with advanced graphics workstations it is impossible to load all available data into main memory – this has proven to be too slow even without texture. We propose to subdivide the area into 16 * 11 square regions of 12.8 km * 12.8 km. The organization of these regions is again done using R-trees (see Table 3).

LoD-Level	DTM grid size	for one region (patch)		projection for entire data set	
		sq. meshes (# points)	texture res. (# pixel)	sq. meshes (# points)	texture res. (# pixels)
0 (full quality)	50 m	256*256	1280*1280	4096*2816	20480*14080
1	200 m	64*64	256*256	1024*704	4096*2816
2	800 m	16*16	64*64	256*176	1024*704
3 (low quality)	3200 m	4*4	64*64	64*44	1024*704

Table 3: Levels of detail for the Styria fly-over

Each square region can be rendered in four levels of details (or not at all). The visualization can start at a point far above ground, from which the entire province is visible. Even with a low-end PC it would be possible to render the model using the lowest level of detail. As the user zooms to one part of Styria, the low quality LoDs for the respective regions are replaced by higher quality LoDs.

This process can be done progressively, e.g. as long as high quality data are not available in main memory, lower resolution data can be used. (The entire model in its lowest resolution can always stay in main memory. Higher resolution patches need to be loaded and removed dynamically.) It is important to do data management in two independent processes. Therefore, the visualization can continue while data are loaded. The process is to a high degree scaleable, e.g. on a high end machine the same frame rates can be achieved at a higher quality.

Conclusions

Large data base contents of entire cities will have a broad appeal for many different groups of users if the data are available via network in an acceptable amount of time and at a high level of quality and realism. To make this scenario become true, fast and intelligent data storage and retrieval concepts need to be implemented. We believe that our approach will contribute to this task. Promising results of the initial test phase have shown that it is possible to handle data of an entire city and prepare them for online visualization and interaction. Looking forward and taking into account that computers become faster and network infrastructure becomes denser, it will be necessary to provide users with high quality data of our environment.

Literature

[1] *Funkhouser T.A., Séquin C.H.:* Adaptive Display Algorithm for Interactive Frame Rates During Visualization of Complex Virtual Environments. In Siggraph Computer Graphics Proceedings, pp. 247-254, 1993.

[2] *Gruber M., Meissl S., Böhm R.:* Das dreidimensionale digitale Stadtmodell Wien. Erfahrungen aus einer Vorstudie. In: VGI (Österreichische Zeitschrift für Vermessung und Geoinformation) 1+2/95, pp. 29-36, 1995.

[3] *Gruber M., Pasko M., Leberl F.:* Geometric versus Texture Detail in 3D Models of Real World Buildings. Proceedings of the Ascona Workshop 95 on Automatic Extraction of Man-Made Objects from Aerial and Space Images. Birkhäuser, Basel 1995.

[4] *Heckbert P.S., Garland M.:* Multiresolution Modeling for Fast Rendering. Proceedings of Graphics Interface '94, Banff, Alberta, Canada, 1994.

[5] *Kofler M., Rehatschek H., Gruber M.:* A Database for a 3D GIS for Urban Environments Supporting Photo-Realistic Visualization. International Archives of Photogrammetry and Remote Sensing, Vol. XXXI, Part B2, Commission III, pp. 198-202, Vienna 1996.

[6] *Lindstrom P., Koller D., Ribarsky W., Hodges L.F., Faust N., Turner G.:* Real-Time Continuous Level of Detail Rendering of Height Fields. In Siggraph Computer Graphics Proceedings, 1996.

[7] *Maciel P.W.C., Shirley P.:* Visual Navigation of Large Environments Using Textured Clusters. ACM Siggraph Symposium on Interactive 3D Graphics, pp. 95-102, 1995.

List of Workshop Participants

Frank Ade, Institute of Communications Technology, Swiss Federal Institute of Technology, ETH Zentrum, Gloriastr. 35, CH-8092 Zurich, Switzerland. Tel./fax: +41 1 632 5280, ++632 1199, ade@vision.ee.ethz.ch

Peggy Agouris, Dept. of Spatial Information Science and Engineering, University of Maine, 5711 Boardman Hall no. 348, Orono, ME 04469-5711, USA. Tel./fax: +1 207 581 2180, ++581 2206, peggy@spatial.maine.edu

Caroline Baillard, IGN, Laborat. Matis, 2, avenue Pasteur, F-94160 St. Mandé, France. Tel./fax: +33 143 98 8436, ++98 8581, baillard@matis.ign.fr

Emmanuel Baltsavias, Institute of Geodesy and Photogrammetry, Swiss Federal Institute of Technology, ETH-Hoenggerberg, CH-8093 Zuerich, Switzerland. Tel./fax: +41 1 633 3042, ++633 1101, manos@geod.ethz.ch

Meir Barzohar, RAFAEL, 12 Lincoln Street, Haifa 34369, Israel. Tel./fax: +9724 825 6027, ++990 6576, mb@lems.brown.edu

Albert Baumgartner, Chair for Photogrammetry and Remote Sensing, Technical University of Munich, Arcisstr. 21, D-80290 Munich, Germany. Tel./fax: +49 89 2892 671, ++89 2809 573, albert@photo.verm.tu-muenchen.de

Frank Bignone, ISTAR, Les espace de Sophia, Bât. F, 80, rte des Lucioles, F-06560 Sophia Antipolis, France. Tel./fax: +33 4 9395 7648, ++9395 8329, bignone@istar.fr

Josef Braun, Carl Zeiss Oberkochen, Carl Zeiss Str. 1, D-73446 Oberkochen, Germany. Tel./fax: +49 7364 204 839, ++204 829, braun@zeiss.de

Claus Brenner, Institute of Photogrammetry, University of Stuttgart, Keplerstr. 11, D-70174 Stuttgart, Germany. Tel./fax: +49 711 121 3398, ++121 3297, claus.brenner@ifp.uni-stuttgart.de

Douglas Climenson, DARPA Consultant, 206 Piney Point Landing, Grasonville, MD 21638, USA. Tel./fax: +1 410 827 8494, ++827 6753, dclimens@shore.intercom.net

John Danahy, Centre for Landscape Research, University of Toronto, 230 College Street, Toronto, Ont., Canada. Tel.: +1 416 978 3551, land@clr.utoronto.ca

Ian Dowman, University College London, Gower Street, London, WC1E 6BT, UK. Tel./fax: +44171 380 7226, ++380 0453, idowman@ps.ucl.ac.uk

Heinrich Ebner, Chair for Photogrammetry and Remote Sensing, Technical University of Munich, Arcisstr. 21, D-80290 Munich, Germany. Tel./fax: +49 89 2892 2671, ++ 2809 573, ebn@photo.verm.tu-muenchen.de

Wolfgang Eckstein, Institut für Informatik, TU München, Orleanstr. 34, D-81667