PANORAMIC SCENES FOR TEXTURE MAPPING OF 3D CITY MODELS

Norbert Haala, Martin Kada

Institute for Photogrammetry (ifp), University of Stuttgart, Germany
Geschwister-Scholl-Strasse 24D, D-70174 Stuttgart
Norbert.Haala@ifp.uni-stuttgart.de

KEY WORDS: visualisation, Virtual Reality, Photo-realism, Terrestrial

ABSTRACT:

Urban models can be collected area covering and efficiently based on aerial data like stereo images or LIDAR. While a large number of applications like simulations are already feasible based on the available three-dimensional building representations, a further quality improvement is required for some other tasks. Especially, if very realistic visualisations from pedestrian viewpoints have to be generated, the quality and amount of detail, which is available for urban models generated from aerial data has to be improved. As an example, due to the viewpoint restrictions of airborne platforms, image texture for the facades of the buildings frequently is not available from airborne sensors. Since this information is crucial to improve the visual appearance of urban models for pedestrian perspectives, alternative data sources have to be applied. Thus, image texture for the faces of existing building models is frequently generated based on manual mapping of terrestrial images. As it will be discussed within the paper, the efficiency of the manual mapping process can be increased considerably by the application of panoramic scenes. If these panoramic scenes are collected from a high-level system based on a rotating CCD line scanner, large areas are covered at high resolution and superb image quality. Thus, the processing of a single scene is sufficient to extract texture for a considerable number of buildings. After the mapping of terrestrial image against the available building model is defined, the process of texture extraction and placement can be realized very efficiently if the functionality of 3D graphics hardware is used. In contrast to the application of standard viewers, this technique allows to model even complex geometric effects like self-occlusion or lens distortion. By these means a very fast and flexible on-the-fly generation of façade texture using real world imagery is feasible.

1. INTRODUCTION

The development of tools for the efficient and area covering collection of 3D city models has been a topic of intense research for the past years. A good overview on the current state-of the-art is for example given in (Baltsavias, Grün, van Gool 2001). Meanwhile, a number of algorithms are available, which are usually based on 3D measurements from airborne stereo imagery or LIDAR. For this reason, the production of virtual city models is more and more becoming a standard task of photogrammetric data collection, resulting in a growing availability of data sets, which include 3D representations of buildings.

Frequently, these 3D models are used to generate photorealistic visualisations of the urban environment. This type of application is for example required in the context of urban planning, tourism or entertainment, like games based on real locations. Airborne data collection is suitable to efficiently provide a complete set of 3D building models, mainly representing the footprints and the roof shapes of all buildings at sufficient detail and accuracy. On the other hand, viewpoint restrictions of airborne platforms frequently limit the amount of information, which can be made available for the facades of the buildings. For this reason, building models collected based on airborne data can only be applied for high quality visualisations if virtual images from elevated viewpoints have to be generated. While this type of images is preferable in order to generate overviews of larger areas, a number of applications in the context of urban planning or 3D navigation require visualisations at very high degree of realism for pedestrian viewpoints. This is only feasible, if data from terrestrial platforms is additionally available during the generation of the urban models. As an example, photorealistic visualisations of building façades require the availability of texture from terrestrial images.

A common task to be solved in this context is the mapping of terrestrial images to the facades of existing 3D building models. Frequently, this is realized by a GUI, which allows a human operator to select corresponding primitives between the available building model and the respective images. As it will be discussed in section 2, this time consuming process of manually mapping can be speed up considerably if panoramic scenes are applied. In our experiments the imagery is collected from the high-level panoramic camera system EYSCAN, developed by the KST GmbH in a cooperation with the German Aerospace Center. By this system, large areas can be depicted at high resolution and superb image quality. Thus, a single scene is sufficient to provide texture for a considerable number of buildings, especially if suitable camera stations have been selected during image collection. Since the number of images to be processed for texture mapping is cut down significantly, the overall effort is reduced, even though manual interaction is still necessary. For geometric processing of the panoramic scenes, the exterior orientation is determined from control points, which are measured manually. Control point information is provided from the available building models, which are depicted in the scene.

If the mapping between building model and image is defined, the visualisation of the textured building models is feasible by standard viewers. Nevertheless, this requires pre-processing steps like the provision of corresponding image and model coordinates, the definition of occluded parts, or the selection of optimal texture if multiple images are available. As it is discussed in section 3 this task can alternatively be solved by programmable graphics hardware. By these means the computa-

tional power of such devices can be used efficiently in order to allow for the on-the-fly generation of façade texture.

2. TEXTURE MAPPING FROM PANORAMIC IMAGES

One general problem for the provision of façade texture from terrestrial images is the large amount of terrestrial scenes to be processed. As an example, within the project "World Models for Mobile Context-Aware Systems" (Stuttgart University 2003) a detailed virtual landscape model of the city of Stuttgart had to be made available. Within the first phase of the project, façade texture was collected for a number of buildings in the historical central area of the city by manual mapping. This manual mapping was based on approximately 5,000 terrestrial images collected by a standard digital camera. In order to extract the façade structures from these images, which were available for approximately 500 buildings several man months were required.

One approach to increase the efficiency of this process is to reduce the number of scenes to be processed for texture mapping by the application of panoramic images. Frequently, these panoramic images are used for tourist purposes in internet applications. For this type of application, the scenes can be generated by the combination of several overlapping images from simple digital cameras. In contrast, for our investigations a high resolution CCD line scanner, which is mounted to a turntable was applied. By these means high quality panoramas can be captured, which allow for precise photogrammetric processing, i.e. in applications like 3D reconstruction or mobile mapping. Especially, if the panoramic scene is collected from an elevated point in order to minimize occlusion, a large area can be covered. Thus, texture for a considerable number of buildings is available from a single scene. In addition to the reduced amount of images to be processed, the application of panoramic scenes allows to minimize changes in illumination, since image collection at different epochs is avoided.

2.1 Data Collection

For our investigations, panoramic scenes were captured by the camera system EYSCAN. An exemplary scene collected from this system is depicted in Figure 1. In order to demonstrate the available resolution for this type of imagery, an enlarged section of the complete scene is additionally presented.

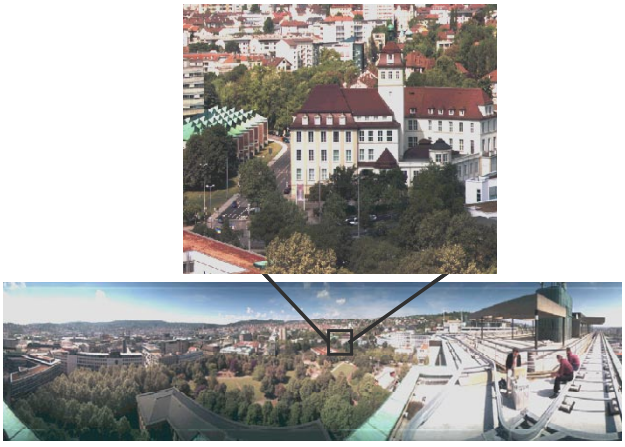

Figure 1: Image collected from panoramic camera EYSCAN.

Figure 2 depicts a 3D view generated from the 3D city model of Stuttgart, which was used as basic dataset within our investigations. The model was collected on behalf of the City Surveying Office of Stuttgart semi-automatically by photogrammetric stereo measurement from images at 1:10000 scale (Wolf 1999). For data collection, the outline of the buildings from the public Automated Real Estate Map (ALK) was additionally used. Thus, a horizontal accuracy in the centimeter level as well as a relatively large amount of detail could be achieved.

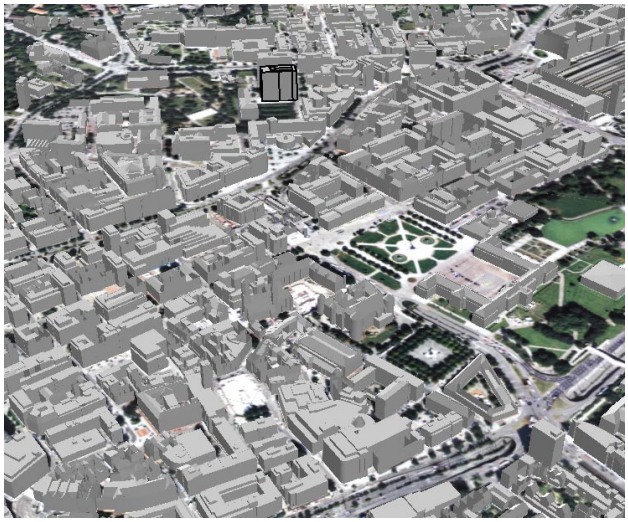

Figure 2: Available 3D urban model.

The resulting model contains the geometry of 36,000 buildings. In addition to the majority of relatively simple buildings in the suburbs, some prominent historic buildings in the city centre are represented in detail by more than 1000 triangles, each. The panoramic image depicted in Figure 1 was collected from the top of the building, which is represented by the black wireframe lines in Figure 2.

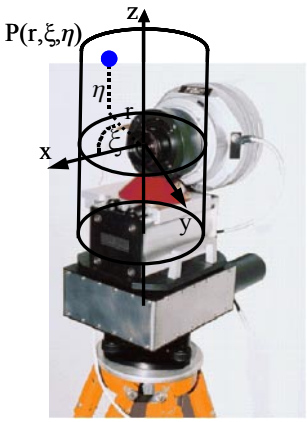

Figure 3: EYSCAN camera with cylindrical coordinate system.

The camera system EYSCAN used within our investigations was developed by the KST GmbH in a cooperation with the German Aerospace Center. The system, which is depicted in Figure 3, is based on CCD line mounted parallel to the rotation axis of a turntable. Thus, the height of the panoramic image is determined by the number of detector elements of the CCD line. In contrast, the width of the image is related to the number of single image lines, which are captured during the rotation of the turntable while collecting the panorama. In our experiments, this resulted in an image height of 10.200 pixels, while during a

360° turn of the camera more than 40.000 columns were captured. Since the CCD is a RGB triplet, true colour images are available after data collection. The spectral resolution of each channel is 14 bit, the focal length of the camera is 60mm, and the pixel size is 7μm.

2.2 Geometric Processing

In order to map the visible faces of the buildings to the respective image patches of the panoramic scene, corresponding image coordinates have to be provided for the 3D object points of the building models. In accordance to the processing of standard perspective images, the object coordinates $\mathbf{X}_P$ are linked to the corresponding camera coordinates $\mathbf{x}$ based on the well known collinearity equation

$$\mathbf{x} = \frac{1}{\lambda}\mathbf{R}^{-1} \cdot \left(\mathbf{X}_P - \mathbf{X}_0\right)$$

which defines a transformation between two Cartesian coordinates system. In accordance to the approach described by (Schneider & Maas 2003), a cylindrical coordinate system is additionally introduced to simplify the transformation of panoramic imagery. In this system, which is overlaid to the picture of the EYESCAN camera in Figure 3, the parameter ξ represents the scan angle of the camera with respect to the first column of the scene. The radius of the image cylinder is given by the parameter r. In the ideal case, this parameter is equal to the principal distance of the camera. The parameter η represents the height of an image point above the xy-plane. Thus, this parameter is related to the vertical distance of the object point to the camera station. The transformation between the cylindrical camera coordinates r, ξ, η and the Cartesian camera coordinates is then given by

$$\mathbf{x} = \begin{bmatrix} x & y & z \end{bmatrix}^T = \begin{bmatrix} r \cdot \cos\xi & -r \cdot \sin\xi & \eta \end{bmatrix}^T$$

In the final step, the transformation between the cylindrical camera coordinates and the pixel coordinates m, n is defined by

$$n = -\frac{\eta + \eta_0}{d_v}, \quad m = \frac{\xi}{d_h}$$

Similar to the processing of frame imagery, the pixel coordinate n in vertical direction is determined by the corresponding component η_0 of the principal point and the vertical resolution d_v, which is defined by the pixel size. In contrast, the horizontal resolution d_v required to compute pixel coordinate m in horizontal direction is defined by the rotation angle of the CCD line per column during collection of the panoramic image.

The required exterior orientation parameters of the scene were computed from a spatial resection. In order to allow for an interactive measurement of the required control points the available 3D building models were approximately mapped to the panoramic image. Based on this approximate mapping, a sufficient number of visible faces, which were distributed over the complete scene were selected and manually measured in the image. In principle, the geometric quality of the EYSCAN camera allows for accuracies of point measurement in the sub-pixel level (Schneider & Maas 2003). Still, in our experiments only accuracies of several pixel could be achieved. In our opinion, this results from the fact, that the fit between model and image after spatial resection is not only influenced by the geometric quality of the image, but also from the accuracy, level of detail and visual quality of the available 3D building models used for the provision of the control points. While our urban model provides a reliable representation of the overall shape of the visible buildings, the amount of detail is limited especially for the facades. As it is discussed earlier, this situation is typical for 3D city model data sets, which are collected from airborne data.

Figure 4: Building models mapped to panoramic image.

Figure 4 depicts the mapping of the available building models against a part of the panoramic image based on the result of spatial resection. The visual quality of virtual images, which can be generated from building models, which are textured using the collected panoramic scene is demonstrated in Figure 5.

Figure 5: Scenes generated using buildings with texture from panoramic image.

3. HARDWARE-BASED FAÇADE TEXTURE EXTRACTION

The virtual image depicted in Figure 5 was generated using a standard VRML viewer. For 3D visualisation, just a link between the 3D object coordinates of the geo-referenced building

models to the corresponding image coordinates was provided., Additionally, the visibility of each building polygon was controlled in advance based on an occlusion detection and an analysis of normal vector of each building polygon.

One problem resulting from the application of such standard tools for visualisation is, that complex transformations can not be realized during texture mapping. Since the façade texture is just interpolated between the given image coordinates of the building polygons, the panoramic image geometry results in straight object lines, which are depicted as curved lines in the texture image. Since for the scenes depicted in Figure 4 and Figure 5 the facades of the buildings only cover small sections of the applied panoramic images, these panoramic distortions can be neglected during the visualisation process. Nevertheless, if image texture is extracted for larger sections, these distortions can be avoided by a suitable a pre-processing step. For this purpose, a 'distortion free' image is generated by mapping the image pixels from the panoramic cylindrical surface onto a tangential plane, which is defined by the respective building face. This step is similar to the rectification of façade imagery, which is preformed if terrestrial images are collected by standard consumer type cameras. In that context, effects of perspective geometry and lens distortion are also eliminated by such a pre-processing step.

As it will be discussed within this section, this task can alternatively solved on-the-fly by programmable graphics hardware. By these means, problems resulting from self-occlusions can additionally be solved and multiple images can be integrated during texture mapping. Another benefit of our approach is that the façade texture is directly extracted from the original images; no intermediate images have to be generated and stored. Additionally, within the whole process image pixels are interpolated only once, which results in façade textures of higher quality.

The approach described in this article is based on technologies that can be found in today's commodity 3D graphics hardware. Graphics processing units (GPU) that are integrated in modern graphics cards are optimized for the transformation of vertices and the processing of pixel data. As they have evolved from a fixed function to a programmable pipeline design, they can now be utilized for various fields of applications. The programs that are executed on the hardware are called shaders. They can be implemented using high level programming languages like HLSL (developed by Microsoft) (Gray, 2003) or C for graphics (developed by NVIDIA) (Fernando and Kilgard, 2003). In our approach shaders are used to realize specialized projective texture lookups, depth buffer algorithms and an on-the-fly removal of lens distortions for calibrated cameras. This approach can be implemented based on the graphics API Direct3D 9.0 which defines dynamic flow control in Pixel Shader 3.0 (Microsoft, 2003). By these means, the lens distortion in the images can be corrected on-the-fly in the pixel shader.

3.1 Texture Extraction and Placement

Our approach uses the graphics rendering pipeline of the graphics card to generate quadrilateral texture images. In general, the function of the pipeline is to render a visualisation of a scene from a given viewpoint based on three-dimensional objects, textures and light sources.

Because the texture images, which are mapped against the façades during visualisation, have to be represented by quadrilaterals, the polygons of the building are substituted by their bounding rectangles during the extraction process. For these bounding rectangles 3D world coordinates are available. This information is used to calculate the corresponding image pixels, which provide the required façade texture. In more detail, the first step is to set up the graphics rendering pipeline to draw the entire target pixel buffer of the final façade texture. For this purpose, the transformation matrices are initialized with the identity, so that drawing a unit square will render all pixels in the target buffer as wanted. As no color information is provided yet, a photograph must be assigned to the pipeline as an input texture from where to take the color information from. As mentioned above, the polygon's projected bounding box defines the pixels to be extracted from the input texture. So in addition to the vertices, the texture coordinates of the four vertices of the unit square are specified as the four-element (homogenous) world space coordinates of the bounding box. Setting the texture transformation matrix with the aforementioned transformation from world to image space concludes the initialization.

Figure 6: Projected 3D building model overlaid on the input photograph

Figure 7: Resulting 3D building model with the extracted textures placed on the façade polygons.

During rendering, the rasterizer of the GPU linearly interpolates the four-dimensional texture coordinates across the quadrilateral. A perspective texture lookup in the pixel shader results in the perspectively correct façade texture (see Figure 6 and Figure 7). After the extraction, the textures need to be placed on the corresponding polygons. In order to find the two-dimensional texture coordinates for the polygon vertices, a function identical to glTexGen (Shreiner, 2003) of OpenGL is used.

3.2 Image Fusion

A common problem is that parts of the building façades are not visible in the photograph due to self-occlusions. If this effect is not modeled correctly erroneous pixels are extracted from the respective texture image (see Figure 9).

Figure 8: Original input image.

Figure 9: Textured building model, occlusion culling disabled.

To avoid such artifacts, invalid pixels that belong to other polygons must be identified and marked. By using the depth buffer algorithm, the closest polygon for each pixel in the photograph can be determined. We use a pixel shader to calculate the depth value and render it directly into a 32 bit floating-point depth texture. During texture extraction, this value is then read out in the extracting pixel shader using the same texture coordinates as for the color lookup. After the perspective divide is applied to the texture coordinates, the z-component holds the depth value for the current polygon pixel. A comparison of these two depth values then determines if the pixel in the color value belongs to the polygon. Figure 10 examplarily shows the result of texture mapping where occluded pixel values have been masked out. When processing more then one photograph, the final color can be merged by using the closest, non-occluded pixel in all images. Even though the approach is brute force, it is still very efficient with hardware support.

If texture image information from various positions is available the final color can be merged by using the closest, non-occluded pixel in all images to generate the final façade texture (Figure 11).

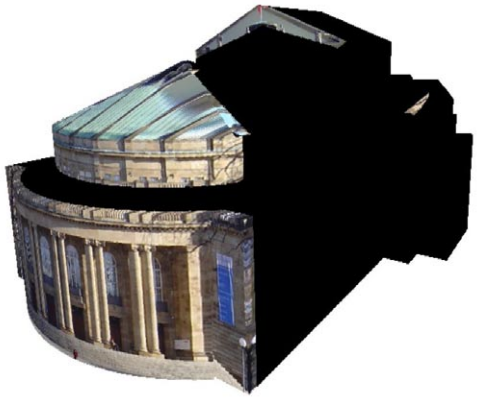

Figure 10: Textured building model, occlusion culling enabled.

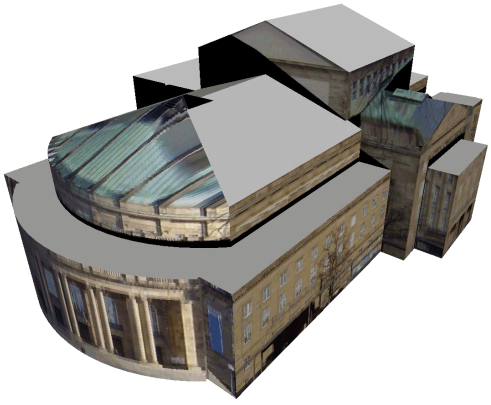

Figure 11. Mapping from multiple images with occlusion free pixels selected.

The texture extraction algorithm was implemented in C++ with the graphics API Direct3D 9.0, which includes the high level shading language HLSL. The performance analysis has been conducted on a standard PC with an Intel 4 3.0 GHz Processor, 1GB of DDR-RAM and a graphics card that is based on the ATI 9800 GPU with 256 MB of graphics memory. With all features enabled, the extraction time for a model of approx. 146 polygons and 16 input images is still below one second.

4. CONCLUSION

If existing urban models, which are frequently available from airborne data collection are used for applications like realistic visualisations of urban scenes from terrestrial viewpoints, a refinement using terrestrial data collection is required. The evaluation of the terrestrial data can be simplified considerably by integrating these existing building models to the respective processing steps. Within the paper, the combined processing was mainly demonstrated for geoereferencing the terrestrial data sets by coregistration to the given models in order to provide image texture. However, this combined processing is even more important if terrestrial data sets are used for geometric improvement of the given models.

Meanwhile, real-time visualisation of virtual 3D city models is integrated in a growing number of applications. This is also triggered by the availability of modern graphic cards, which allow high-end visualisation using reasonably priced standard hardware. As it has been demonstrated in the second part of the paper, the great computational power of these systems can be

used even more effeciently by programmable graphics hardware. This allows for a direct implementation of relatively complex algorithms like texture mapping of building façades using geo-referenced terrestrial images. By these means 'photogrammetric' tasks like the transformation and mapping of world to image coordinates – also for complex camera geometries - are directly integrated in the graphics rendering pipeline in order to allow for a time efficient solution. This demonstrates the potential of integrating techniques from computer graphic and photogrammetry for time critical applications like the link of virtual 3D models and terrestrial imagery for real-time visualisation of textured city models.

5. REFERENCES

Baltsavias, E., Grün, A. & van Gool, L. [2001]. *Automatic Extraction of Man-Made Objects From Aerial and Space Images (III)*. A.A. Balkema Publishers

Fernando, R. and Kilgard, M. (2003). *The Cg Tutorial*. Addison-Wesley.

Gray, K. (2003). *The Microsoft DirectX 9 Programmable Graphics Pipeline*. Microsoft Press.

Microsoft, 2003. DirectX Documentation for C++. Microsoft DirectX 9.0 SDK. http://msdn.microsoft.com/library/default.asp?url=/downloads/list/directx.asp

Schneider, D. & Maas, H.-G. [2003]. Geometric Modelling and Calibration of a High Resolution Panoramic Camera. Optical 3-D Measurement Techniques VI. Vol. II, pp.122-129.

Shreiner, D., Woo, M. and Neider, J. (2003). *OpenGL Programming Guide* (Version 1.4). Addison-Wesley.

Stuttgart University [2003]. Nexus World Models for Mobile Context-Based Systems. http://www.nexus.uni-stuttgart.de/.

Wolf, M. [1999]. Photogrammetric Data Capture and Calculation for 3D City Models. Photogrammetric Week '99, pp.305-312.