

See discussions, stats, and author profiles for this publication at: <http://www.researchgate.net/publication/228639937>

# A new texture mapping algorithm for photorealistic reconstruction of 3D objects

ARTICLE · OCTOBER 2009

---

CITATIONS

5

---

DOWNLOADS

7

---

VIEWS

35

## 1 AUTHOR:



[Thomas Hanusch](#)

ETH Zurich

6 PUBLICATIONS 111 CITATIONS

SEE PROFILE

# A NEW TEXTURE MAPPING ALGORITHM FOR PHOTOREALISTIC RECONSTRUCTION OF 3D OBJECTS

T. Hanusch

Institute of Geodesy and Photogrammetry, Swiss Federal Institute of Technology (ETH) Zurich, ETH-Hoenggerberg, CH-8093, Zurich, Switzerland - thomas.hanusch@geod.baug.ethz.ch

Commission V, WG V/4

**KEY WORDS:** Texture, Photo-Realism, Colour, Rendering, Visualization, Resolution, Cultural Heritage, Virtual Reality

## ABSTRACT:

In the last decade, new developments in remote sensing, photogrammetry and laser scanning generated new possibilities for acquisition, processing and presentation of different objects at various scales. Beside object modelling, photorealistic texturing is still a challenging task. This paper focuses on a new texture mapping technique, which extends the known 2.5D approaches of texture mapping to 3D objects. A new algorithm will be described, suitable to achieve high quality results by eliminating problems of current approaches e.g. resolution dependency, need of huge amount of memory and handling of colour and brightness differences. A new, vector based visibility analysis, using vector algebra instead of pixel based algorithms will be presented. In the context of texture mapping, the fusion of different texture sources (different images) with different illumination and exposure conditions as well as images with different resolutions is an important task. To achieve best results in terms of homogeneity and level of detail, a multi-image texturing approach was developed, to preserve the fine texture information of the object. To adjust the brightness levels of the image dataset, a global and a local colour adjustment was developed. The software "Blender" was used to generate high quality images, movies and animations. The result of the investigations and developments is a software system to texture 3D models photorealistically and automatically. All algorithms are discussed in detail. The performance is demonstrated using artificial and real world data sets.

## 1. INTRODUCTION

Textured models are established as a proper possibility to present the results of photogrammetric data acquisition and processing to experts as well as to non-experts. The expectations of the customers are increasing according to the development of the hardware. It started from simple visualizations of small 2.5D areas over increased datasets with artificial texture e.g. colour coded surface models to photorealistic textured 2.5D models. Now, the step towards 3D objects with millions of points, photorealistic texture and real-time visualizations is in progress. This paper describes a whole workflow from the oriented images and the geometry information over the visibility analysis and the global and local brightness correction of the images up to the final visualization using open source software. Therefore, a number of algorithms were evaluated, modified, developed and combined.

For texture mapping of 3D objects a visibility analysis has to be conducted. The approaches developed for 2.5D surfaces or for true-orthophoto generation (Rau, 2002), (Amhar, 1998), (Zhou, 2003), (Biasion, 2004) have to be adapted to 3D data. The existing 3D algorithms were basically developed for computer sciences applications and have some disadvantages, e.g. high expectations concerning image overlaps, high number of images and short image acquisition times to avoid changes of the illumination conditions. These requirements are mostly problematic to fulfil, especially for photogrammetric data sets. Therefore we developed a new visibility algorithm.

For the triangle to image assignment, existing approaches were slightly modified and adapted (Niem, 1995), (Rocchini, 1999). According to the high sensitivity of the human eye, which

recognizes grey value difference of less than one percent, a colour adjustment is necessary to achieve satisfying results during multi image texture mapping. Different algorithms were presented in (Agathos, 2003) and (Bannai, 2004). In this paper a solution, combining three different processing steps, is presented.

Finally, open source software was used to generate high resolution images, virtual reality models and movies of the processed data.

## 2. DATA ACQUISITION

Two datasets are used to show the performance of the developed algorithms.

- A Khmer head, carved out of sandstone in the 12th or early 13th century, with a height of about 30cm. The geometry as well as the images were acquired in the Museum Riedberg, Zurich, Switzerland (Akca, 2007).
- A part of the St. Gallen Globus, manufactured around 1595 in St. Gallen, Switzerland, with a height of around 230cm, made of wood, fully painted with the earth and sky map of Mercator and diverse other ornaments. The used part is a support of the sphere with a size of 70 x 25 square centimetres. The dataset consists of images acquired using a standard still video camera.

Dataset 1 was used to show the whole workflow beginning from raw-data to the final textured model. Dataset 2 was used to show the brightness correction algorithm in detail.

The following information is necessary and has to be provided to use the presented algorithms:

- Image data
- Interior orientation for each camera with variable settings, e.g. focus
- Sensor-model
- Exterior orientation of each image
- Elevation- or 3D surface model as a triangulated irregular network (TIN).

### 2.1 Geometry

The geometry acquisition for the dataset 1 was done using the Breuckmann OptoTOP-SE fringe projection system. 18 point clouds were generated. The following registration was done using the software LS3D (Gruen, 2005). The final geometric model used for texture mapping was generated using PolyWorks and consists of 600k triangles. In (Akca, 2007) acquisition and modelling are described in detail.

For the second dataset, no additional geometric information was generated. The few needed object points are acquired using the oriented images and manual measurements in the software package PhotoModeler Pro 5 (Eos Systems).

### 2.2 Image Data

The texture of the Khmer head was acquired using the standard still video camera Sony DSC-W30 with 6 mega pixel, in a circle around the object. A professional illumination system consisting of two diffuse lights was used, to reduce the radiometric differences between the images and shadow effects at the complex parts and object silhouettes. The interior and exterior orientations were computed using a photogrammetric bundle adjustment with self-calibration (Akca, 2007).

The images of the Globe dataset were acquired using a standard still video camera Sony F828 with a focal length of around 7mm. For the illumination a semi-professional light system was used. Nevertheless, because of other poor acquisition conditions the images were very inhomogeneous. Therefore this dataset is suitable to show the performance of the brightness correction procedure. To cover the object with a size of 70 x 25 square centimetres, four overlapping images are necessary. For our tests only two images were used. The orientation and camera calibration was done in the software PhotoModeler using manual measurements.

## 3. ALGORITHM

### 3.1 General workflow

Figure 1 shows the complete workflow. Part I gives an overview about the necessary data. Part II covers the developed algorithms and will be described in detail.

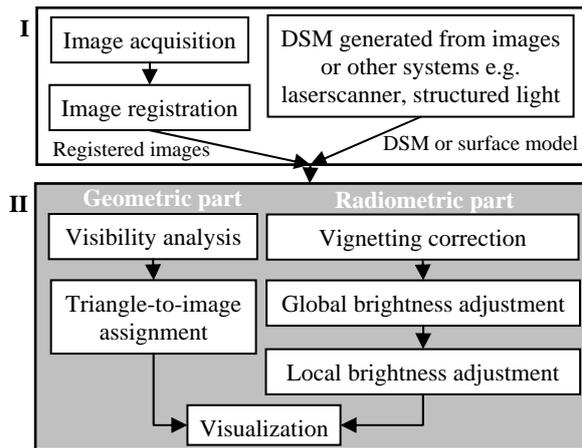


Figure 1. Texture mapping workflow

### 3.2 Visibility analysis

The visibility algorithm was designed to work with unsorted, unclosed and "un-oriented" triangle meshes. This means, the dataset of the elevation or 3D model can be available in any unsorted form. Information about connected triangles is not needed and holes in the mesh do not influence the procedure. "Un-oriented" implies that the usual order of the vertices of each triangle to define front and backside (counter- or clockwise) must not be assumed. Furthermore, the algorithm is generic in a sense, that it can be applied to satellite-, aerial- and terrestrial images. These limited prerequisites enable the handling of automatically generated datasets with a minimum of manual post-processing. The only constraint is that the triangles have no intersections.

The result of the visibility analysis consist of two lists, one contains only fully visible triangles, the other one fully occluded triangles. To use the result of this algorithm in other applications and to minimize the efforts in further processing steps, partly occluded triangles have to be eliminated.

After the import of elevation data, exterior and interior orientation parameters of every camera position, the algorithm conducts the visibility analysis without manual interaction using the following algorithm.

Each triangle is compared with each other to find potentially occluded triangles. Therefore, the vertices of both triangles are projected into image space, in addition the distances between the triangles and the projection centre are calculated. The processing always considers the triangle with the longer distance to the projection centre as the possibly occluded one (in the following named as A) and the triangle with the shorter distance as the possibly occluding (in the following named as B). If all the vertices of A are inside of B, the triangle A is fully occluded by the triangle B. In contrast, if during the whole processing triangle A is never occluded by any other triangle, A is fully visible. Beside these two possibilities, the third possibility, the partial occlusion of triangles, has to be considered. As mentioned before, the results of the visibility analysis should be either fully visible or fully occluded triangles, partially occluded triangles have to be subdivided into these two categories. Therefore, possibly existing intersections of the triangle boundaries in image space are determined. After calculation of these intersecting image points and the equivalent object point in the object space at A, a re-triangulation is done

in the object space using the original vertices, the intersecting points and the points of triangle B which are inside of triangle A. A Delaunay triangulation algorithm with an additional constraint to preserve the original border of the triangle A is applied. The subdivision of triangle A replaces the original triangle A in the dataset. In the following the new triangles are handled like the original triangle. After the whole processing all the triangles are sorted or subdivided in fully visible and fully invisible triangles. Figure 2 shows the basic principle of the visibility analysis algorithm.

During the implementation different special degeneration cases like identical points, lines as well as triangles (in object and in image space) had to be considered to enable a fully automatic procedure.

The algorithm has a complexity of  $O^2$ . To reduce the processing time, the base dataset is reduced by determination of the visible part of the original dataset in each image using the projection of the image boundaries into the object space. In case of an ordered triangle vertex definition, an additional back-face elimination can be conducted for a further reduction of triangles. It has to be pointed out that this step is not compulsory. Nevertheless, it can reduce the amount of processing time significantly, depending on the dataset.

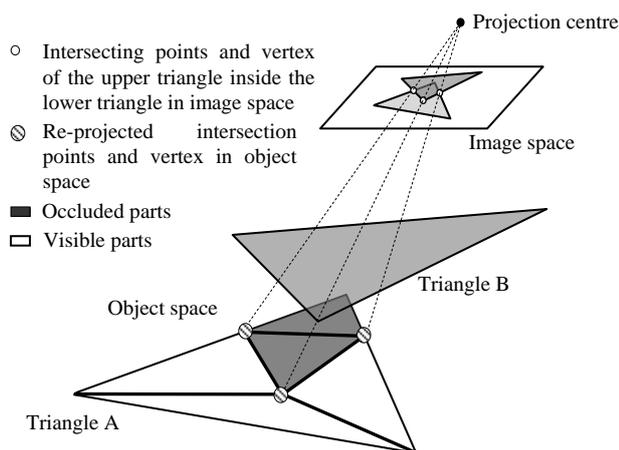


Figure 2. Visibility Analysis

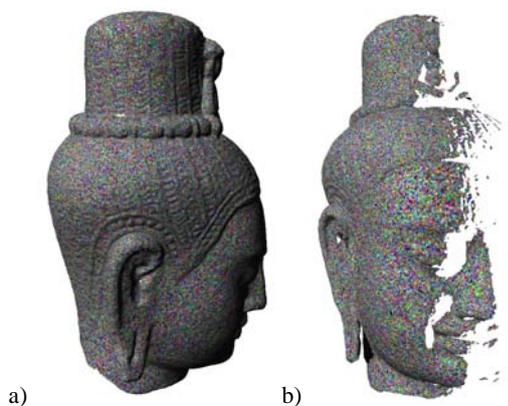


Figure 3. Result of the visibility analysis (image 8)  
 (a) View from camera position  
 (b) Side view

The algorithm requires a small amount of memory. It only stores the triangles in the RAM with 76 byte per point plus 24 byte for each triangle. For example, the required memory for a closed surface of 1 million triangles is around 95 MB.

Figure 3 shows an example of the result of the visibility analysis of the dataset. Corresponding datasets are available for all camera positions. The colours of the triangles in this visualization are defined random wise in order to make it easy to recognize single triangles and to visualize the complexity of the TIN.

### 3.3 Triangle to Image Assignment (TtIA)

The Triangle to Image Assignment combines the separate lists of visible triangles together to achieve a complete textured model. In general, two ways are available to combine multiple data or images.

The first is to combine all available or suitable texture datasets using an averaging procedure for each pixel. The result of this algorithm is a model with a very smooth visual impression. Colour and brightness differences are eliminated. On the other hand this smoothing eliminates the high frequency information and introduces errors, caused by erroneous image orientations (El-Hakim, 2003).

The second method avoids averaging to reduce the loss of high frequency information or details as much as possible. Therefore only one data source for each triangle is used. Consequently high frequent information is preserved. This method is used in the presented workflow. To determine the best texture for each triangle, parameters to evaluate the quality of the texture have to be introduced (Frueh, 2004). The two main parameters, distance and viewing direction, were used for our datasets. The distance is defined by the distance between the triangle and the projection centre of the camera. The angle is defined by the viewing direction of the camera and the surface normal of the triangle. The combination of both parameters finally leads to the resolution of the image data on the object.

For each triangle, lists of available data sources (images) are generated using the results of the visibility analysis. Afterwards the above mentioned texture quality parameters are calculated for each triangle to find the source with the highest resolution. After processing of each triangle, the TtIA is finished. Finally, the texture (uv-) coordinates for each vertex are determined by re-projecting the object coordinates onto the particular image. Figure 4 shows the result of this procedure for two camera positions.

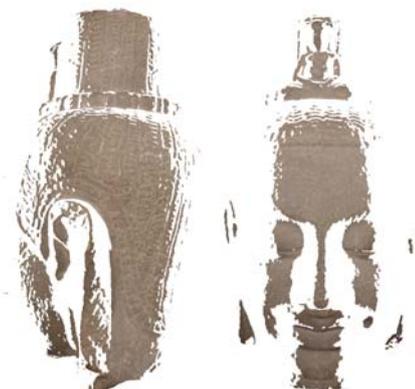


Figure 4. Result of the TtIA for two camera positions

The described algorithm works independently for each triangle. The result of the TtIA is independent of the assignment of the adjacent triangles. Depending on the illumination conditions during the acquisition, the brightness levels of the used images are very inhomogeneous; the result of the textured model using the original images and the TtIA can be speckled. This problem is reduced or eliminated using the following global and local image enhancement steps (chapter 3.4).

### **3.4 Global Grey Value Adjustment**

#### **3.4.1 Colour space**

According to the image acquisition situation with controlled illumination conditions and using one camera with constant settings, we assume the colour values between two adjacent images and the same object points as approximately constant, but according to different acquisition positions and different exposure times we assume the brightness levels of the images as different. Therefore, we apply the following vignetting correction as well as the global correction (chapter 3.4.2-3.4.4) only to the brightness of the images; the colour values are not modified.

To do this independently from the colour and in relation to the human colour and brightness perception, the CIE 1976 ( $L^*$ ,  $a^*$ ,  $b^*$ ) colour space (CIELAB) is used. This colour space was developed by the CIE (Commission Internationale d'Éclairage) to describe all the colours visible to the human eye and to approximate human vision. It aspires to perceptual uniformity, and the L component closely matches human perception of brightness. It can be used to make accurate colour balance or to adjust the lightness contrast using the L component. These transformations are difficult or impossible to handle in the RGB space, because RGB models the output of physical devices, rather than human visual perception. Uniform changes of components in the  $L^*a^*b^*$  colour model aim to correspond to uniform changes in perceived colour. Therefore, the images are transferred from the RGB- to the  $L^*a^*b^*$ - colour space to process only the L-channel (luminance).

#### **3.4.2 Removal of vignetting**

The first step to enhance images is the reduction of the vignetting, also called light falloff effect. Therefore, the  $\cos^4$  relation is used (Hasler, 2004), to reduce the effect significantly, as long as this influence of the lens system can be described sufficiently accurate. For more complex lens systems the approach is not sufficient anymore; more complex algorithms have to be used (d'Angelo, 2004; Goldman, 2005; Litvinov, 2005).

#### **3.4.3 Global brightness correction**

The algorithm calculates one global brightness difference using an averaging over the whole image. This difference value is applied to all pixels of the image. The brightness differences are calculated in the following way. The images are transferred into the  $L^*a^*b^*$  colour space and the L (brightness) component for each common object point is calculated in two images. The difference of these two values is calculated for all points. The global brightness difference is the average value of all the brightness differences. The key factor to achieve acceptable results is the accurate orientation of the images, otherwise the result of the grey value difference calculation is not meaningful, because the grey values of the images are taken from different image positions, which are not related to each other. To reduce this influence as much as possible, two steps of processing were implemented. First, the brightness values are determined using

an average of an area, not only of a single pixel. To calculate the grey values for each point we use a square neighbourhood around the calculated point. Empirical tests have shown that the optimal size of the neighbourhood depends on the dataset and on the quality of image orientation. Best results were achieved using a surrounding of 3x3 up to 7x7 pixels around the calculated image coordinates.

The second part is the improvement of the image registration or orientation itself. Two potential methods are existing. First, an improvement of the whole image orientation. It covers the exterior as well as the interior orientation including distortion parameters. According to the fact, that the here presented algorithm should work with different camera models and with different definitions of distortion parameters, the approach to implement each camera model is not flexible enough. Therefore, a second and more flexible procedure was defined to match the single points in two images to each other. At the beginning, one image is set as master image, the image point coordinates are calculated for all images using the given orientation parameters and the camera model. To improve the position of each point in the second (slave) image, a cross correlation matching is conducted. During this step, the position of the point in the slave image can move up to four pixels. The limit of four pixels was used to avoid big shifts in "uncooperative" regions, e.g. regions without sufficient texture and contrast. The algorithm was implemented to achieve sub-pixel accuracy and works in four steps. First the extraction of an 11-by-11 template around the master control point and a 21-by-21 region around the point in the slave image is conducted, followed by the calculation of the normalized cross-correlation values between template and the selected search region. The detected absolute peak of the cross-correlation matrix is used to adjust the coordinates of the point in the second image.

The global brightness correction improves the brightness level over the whole image. Local differences, for example caused by changing the illumination source- and camera-position cannot be modelled in this way. To evaluate the meaningfulness and the usability of the global brightness correction the standard deviation of the brightness differences are introduced. If the standard deviation exceeds a defined threshold, e.g. half of the average value of the brightness differences, the grey value difference between the images can not be approximated with one value anymore.

#### **3.4.4 Local brightness correction**

To handle non-uniform brightness differences between the images, a local brightness correction has to be applied. The grey value differences are calculated in each point of the images, like in the approach before (chapter 3.4.3). In contrast to the global correction, these differences are not combined to an average value. The differences are associated with the image coordinates of the points. This point cloud of brightness differences can be interpreted as TIN. Using these initial points, a brightness difference grid can be interpolated over the whole image (Figures 6 and 8). The first test to apply this procedure was done using a simple inverse distance weighting function using the closest three points. The key factors for satisfying results are the point density and the accuracy of the single brightness values. The accuracy primary depends, like in the global algorithm, on the correct orientation of the images and on the back-projection of the points. This problem was solved using the matching procedure mentioned before. To check the behaviour with sparse datasets, we used the Globe dataset with a point density of about one point per five square centimetres.

The result of the inverse distance weighting is shown under [www.photogrammetry.ethz.ch/~hanuscht/tm\\_beijing.html](http://www.photogrammetry.ethz.ch/~hanuscht/tm_beijing.html).

It shows that the interpolation of the difference surface using this method is not sufficient, especially not smooth enough. Therefore, a more complex function was introduced. Figure 5 was generated using an artificial dataset. The original image A was modified in a way that the brightness increased in vertical direction (Figure 5 a, side A'), starting from the middle of the image. The generated image was corrected using the described algorithm (Figure 5 b).

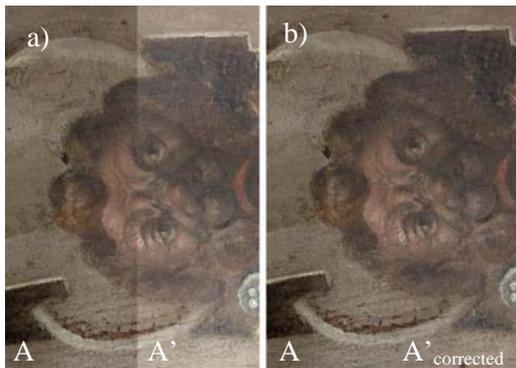


Figure 5. Correction of the artificially modified image  
Original image A and artificially modified image A'  
Original image A and corrected image A'<sub>corrected</sub>

The used interpolation method is a biharmonic spline interpolation (Sandwell, 1987). For each pixel of the slave image, a correction value is calculated and added to the original grey value. Figure 6 shows the interpolated grey-value-difference-surface for the artificially modified dataset.

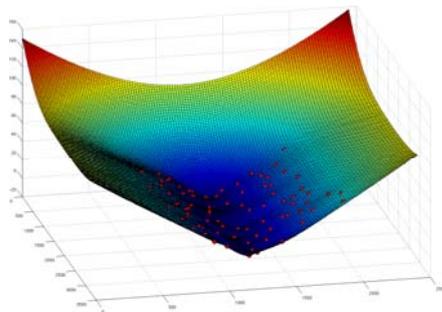


Figure 6. Grey difference surface interpolated using a biharmonic spline function and used points (red)

The grid space of the generated surface is equal to the pixel size. Due to this, for each pixel of the image a correction value is provided to change the brightness according to the master image. It has to be considered, that the regions outside the used point cloud are extrapolated. This does not influence the result, because the main part of the object is covered with common points. If this is not the case, the usage of the correction values should be restricted to defined areas.

Figure 7 shows the result of the whole algorithm using real data, without artificial brightness modifications. The visible seams (a) are significantly removed (b).

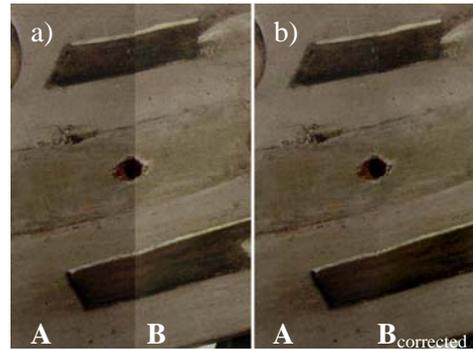


Figure 7. Comparison between:  
a) Original data, image A (left) and image B (right)  
b) Original image A (left) corrected image B (right)

Figure 8 shows the brightness difference surface. After the correction of the image brightness, the visualization can be conducted.

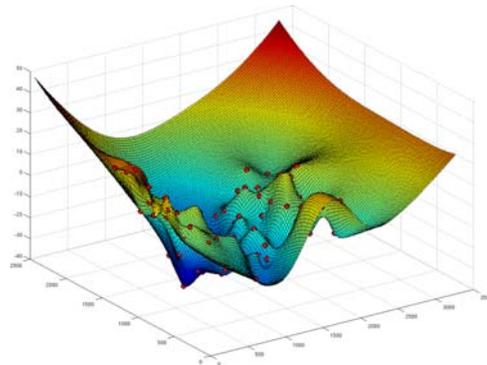


Figure 8. Grey difference surface for the real data example

#### 4. VISUALIZATION

To visualize the extracted data and to achieve a satisfying result the open source software system Blender was used (Blender, 2008). It is an open source 3D content creation suite, available for all major operating systems under the GNU General Public License. Among other features it can be used for modelling, UV unwrapping, texture mapping, animation and rendering, comparable with a commercial system, e.g. Maya (Autodesk). To fulfil our requirements, only a small part of the functionality was used. The supported programming language Python was used to integrate this software into the existing workflow. Therefore, a script was generated to import the triangle and coordinate lists from the triangle to image assignment. The material for each triangle was defined by the calculated texture coordinates (u-,v- coordinates) (chapter 3.3). Using this information the object can be visualized inside the software in real-time and virtual reality models can be exported in different formats (VRML, COLLADA, etc.). A further step is the generation of movies and high resolution images. Therefore, different steps are necessary. For example different illumination sources and camera positions, as well as the camera path for the camera movement has to be defined in an iterative procedure. These steps require, depending on the experiences of the operator, some time to become familiar with the functionality of the software. Finally movies and images in any resolution and from any position can be generated and exported. The bottleneck of this procedure is the handling of large datasets in

Blender. At the beginning, the whole dataset with 600k triangles was processed. With this number of triangles, the performance of Blender was very poor. Nevertheless, a VRML model was exported with the full number of triangles (600k). The quality was satisfying, but real time navigation was more or less impossible. Due to this, a reduced dataset of 100k triangles was generated. With this dataset, the performance of the real time visualization was acceptable. The comparison of the visual impression, especially of the visible details, showed that the two datasets are equivalent, because most of the information for the human eye is provided by the texture, not by the geometry. This leads to the fact, that for pure visualization the geometry can be reduced down to a certain level, as long as the full texture resolution is still used. Finally, a virtual flight around the object was produced with the full dataset and a resolution of 1024 x 768 pixels. Figure 9 shows the result of the Khmer head dataset. It was generated using 12 images.



Figure 9. Final result, the textured Khmer head

## 5. CONCLUSIONS

The presented workflow of texture mapping contains all the important steps, from the given orientation data and the original images to the final textured 3D model. It includes a new visibility analysis algorithm fully based on vector algebra. This leads to an image-resolution independent analysis, suitable to handle sparse and dense datasets at the same time, in a fully automatic way without manual interaction. The following Triangle to Image Assignment procedure selects the best texture source for each triangle from multiple images. No averaging is done, this preserves the high frequent texture information. The finest details of the images are preserved. To achieve a photorealistic and seamless textured model, three image enhancement steps were implemented. First, the vignetting was removed using a simple  $\cos^4$  relation. A second step removed the global brightness difference over the whole image, caused for example by different exposure times. In a last step, a local brightness correction was conducted to compensate influences of different or moving light sources or spot light effects by generating a brightness difference surface over the whole image. This surface was interpolated using a biharmonic spline function. Common points in two images are used as input.

The final visualization of the data was done using the open source software system Blender, which provides the possibility for high resolution rendering of single images as well as for the generation of movies.

## 6. FURTHER WORKS

The described algorithms are working well and achieve satisfying results. Nevertheless, some parts of the algorithms will be improved in the future. First, the performance of the visibility algorithm concerning processing time should be improved. A possible solution is the splitting of the dataset in tiles and usage of multi core CPU systems.

Concerning the Triangle to Image Assignment, a patch growing algorithm will be implemented, to reduce the length of borderlines between different texture sources. This will reduce the number of potential seams.

Furthermore, the algorithm of vignetting reduction can be enhanced in order to model more complex lens systems.

A last improvement could be the exchange of the cross correlation function by a least squares matching procedure to improve the point fitting for images with larger angle between the viewing directions.

## ACKNOWLEDGEMENT

The author thanks the Swiss National Science Foundation (SNF) for the financial support to make this work possible.

## REFERENCES

- Akca, D., Remondino, F., Novk, D., Hanusch, T., Schrotter, G., and Gruen, A., 2007. Performance evaluation of a coded structured light system for cultural heritage applications. Videometrics IX, Proc. of SPIE-IS&T Electronic Imaging, San Jose (California), USA, January 29-30, SPIE vol. 6491, pp. 64910V-1-12.
- Amhar, F., 1998. The Generation of True Orthophotos Using a 3D Building Model in Conjunction with a Conventional DTM. IAPRS, Vol. 32, Part 4 "GIS-Between Vision and Applications", pp. 16-22.
- Biasion, A., Dequal, S., Lingua, A., 2004. A new Procedure for the Automatic Production of True Orthophotos. ISPRS Conference proceedings, Istanbul, Commission IV, pp. 538-543.
- Blender, 2008. <http://blender.org>, (accessed 20. Jan. 2008)
- d'Angelo, Pablo, March 21st-24th 2007. Radiometric alignment and vignetting calibration, Workshop: Camera Calibration Methods for Computer Vision Systems - CCMVS2007, Bielefeld University, Germany.
- El-Hakim, S., Gonzo, L., Picard, M., Girardi, S., Simoni, A., 2003. Visualisation of Frescoed Surface: Buonconsiglio Castle - Aquila Tower, Cycle of Months. Proceeding of International Workshop on Visualisation and Animation of Reality-based 3D Models, Tarasp-Vulpera, Switzerland.
- Frueh, C., Sammon, R., Zakhor, A., 2004. Automated Texture Mapping of 3D City Models with Oblique Aerial Imagery. 2nd International Symposium on 3D Data Processing, Visualisation and Transmission (3DPVT'04), Thessaloniki, Greece, pp 275-282.

- Goldman, Dan B. and Jiun-Hung Chen, Okt. 2005. Vignette and Exposure Calibration and Compensation. In Proceedings of ICCV 2005, Beijing, October 2005, pp. 899-906.
- Gruen, A., Akca, D., 2005. Least squares 3D surface and curve matching. ISPRS Journal of Photogrammetry and Remote Sensing, 59 (3), 151-174.
- Hasler D., Süssstrunk S., 2004. Mapping colour in image stitching applications, Journal of Visual Communication and Image Representation, Vol. 15, Nr. 12, pp. 65-90, 2004.
- Litvinov A., Schechner Y. Y., 2005. Radiometric framework for image mosaicking, Journal of the Optical Society of America - A, Vol. 22 , No. 5 , pp. 839-848 (2005).
- Niem, W., Broszio H., 1995. Mapping Texture From Multiple Camera Views onto 3D-Object Models for Computer Animation. Proceedings of the International Workshop on Stereoscopic and Three Dimensional Imaging, Santorini, Greece.
- Rau, J.-Y., Chen, N.-Y., Chen, L.-C., 2002. True Orthophoto Generation of Built-Up Areas Using Multi-View Images. Photogrammetric Engineering & Remote Sensing, Vol. 68, No.6, pp. 581-588.
- Rocchini, C., Cignoni, P., Montani, C., 1999. Multiple Texture Stitching and Blending on 3D Objects. Eurographics Rendering Workshop, Granada, Spain.
- Sandwell, David T., 1987, Biharmonic Spline Interpolation of GEOS-3 and SEASAT Altimeter Data, Geophysical Research Letters, 14, 2, 139-142, 1987.
- Zhou, G., Qin, Z., Kaufmann, P., Rand, J., 2003. Large-Scale City True Orthophoto Mapping for Urban GIS Application. ASPRS Annual Conference Proceedings, Anchorage, Alaska.

