PHOTOREALISTIC OBJECT RECONSTRUCTION USING COLOR IMAGE MATCHING

Yasemin Kuzu

Photogrammetry and Cartography
Technical University of Berlin
Str. des 17 Juni 135, EB 9
D-10623 Berlin, Germany
kuzu@fpk.tu-berlin.de

Commission V, WG V/2

KEYWORDS: Photo-realism, Scene, Reconstruction, Color, Matching, Correlation, Calibration.

ABSTRACT:

In this paper we present a volumetric object reconstruction technique, based on voxel models. Our goal is to create a photorealistic model, keeping the system requirements as low as possible. We used a standard CCD-video-camera which can also capture still images in high resolution. A circular camera setup was done, which was actually accomplished by rotating the object itself, while the camera position was stationary. The object is reconstructed using several steps of processing. The first step results in an approximate model called the visual hull which is the bounding volume of the original object. However, before starting the reconstruction, the system configuration has to be defined. First of all, we need to do a camera calibration. Next, all the acquired images have to be oriented. Therefore, after computing some control points on the object in a local coordinate system, a bundle block adjustment was calculated, including all the images. Since the system setup is mathematically described, the actual model reconstruction can be done. Although the object's visual hull is a good approximation, it does not fully recover concavities of the object. One approach to refine the model, described in this paper uses the object's texture to find corresponding points in the images. With the previously calculated image orientation, the point's three dimensional coordinates can be described and non-surface voxels are carved away. A way to find corresponding points is to perform an area based image matching, i.e. with the normalized cross correlation. Experimental results show that the final set of voxels accurately represents the original object.

KURZFASSUNG:

In dieser Arbeit stellen wir eine Technik zur volumetrischen Objektrekonstruktion vor, die auf einem Voxel-Modell basiert. Das Ziel ist die Erstellung eines photorealistischen Modells mit möglichst geringen Systemanforderungen. Wir haben eine handelsübliche CCD-Videokamera verwendet, welche die Funktion bietet Einzelbilder mit hoher Auflösung zu erfassen. Eine Kreisförmige Kameraanordnung wurde dadurch erreicht, dass das Modell selbst gedreht wurde, während die Kamera nicht bewegt wurde. Das Objekt wird in mehreren Schritten rekonstruiert. Der erste Schritt liefert dabei ein ungefähres Modell, die umgebende Hülle, die das umgebende Volumen des Ausgangsobjektes repräsentiert. Bevor jedoch die Rekonstruktion stattfinden kann, muss die Systemanordnung definiert werden. Dabei muss zuerst die Kamera kalibriert werden. Danach müssen alle aufgenommen Bilder orientiert werden. Nachdem auf dem Objekt einige Passpunkte in einem lokalen System definiert worden sind, wurden alle aufgenommenen Bilder einer Bündelblockausgleichung unterzogen. Da nun die Aufnahmeanordnung mathematisch beschrieben werden kann, wurde die eigentliche Objektrekonstruktion durchgeführt. Trotzdem die umgebende Hülle eine gute Näherung ist, wird sie konkave Regionen des Objektes nicht richtig erfassen. Eine Vorgehensweise das Modell zu verfeinern verwendet die Textur des Objektes um korrespondierende Punkte in den Bildern zu finden. Mit den zuvor berechneten Bildorientierungen können nun die Raumkoordinaten des Punktes beschrieben und nicht-Oberflächen Voxel entfernt werden. Ein Weg, korrespondierende Punkte zu finden ist die Durchführung einer normierten Kreuzkorrelation. Experimentelle Ergebnisse zeigen, dass das endgültige Modell das Ausgangsobjekt zuverlässig repräsentiert.

1. INTRODUCTION

A fundamental problem in photogrammetry and computer vision is the reconstructing of the shape of 3D objects from photographs. The manual creation of highly detailed real objects is complex and time consuming. In this paper, we describe an efficient image-based approach to compute volume models from color images.

In many fields like automatic 3D model reconstruction, virtual reality, CAD/CAM, robotics and entertainment there is a need for developing photorealistic models of real environments that look and move realistically.

Volumetric representations use voxels to model the scene, which consume large amounts of memory, for example 256^3 bytes (16.77 mbytes) for a rather small cube (256 units in each direction). However, with the rapid advances in hardware this becomes less of a problem and volumetric representations become more attractive.

The model of the 3D object can be easily acquired by volume intersection methods. These methods are often referred to as shape from silhouette algorithms. The intersection of silhouette cones from multiple images defines estimate geometry of the object called the visual hull (Szeliski, 1991), (Niem, 1994), (Kutulakos and Seitz, 1998) (Vedula et al., 1998) and (Kuzu and Rodehorst, 2001). This technique gives a good approximation of the model. However, the concavities on an

object cannot be recovered since the viewing region doesn't completely surround the object. The recent attempts are based on voxel coloring algorithms (Seitz and Dyer, 1997), (Culbertson et al, 1999), (Kuzu and Sinram, 2002). These algorithms use color consistency to distinguish surface points from the other points in the scene making use of the fact that surface points in a scene project into consistent (similar) colors in the input images. In this paper we are using a color image matching algorithm to refine the visual hull of the object.
Prior to the object reconstruction, the camera was calibrated, as it is described in chapter 2.1. Some control points on the object itself were defined in a local coordinate system, which were used to perform a bundle block adjustment with all 22 images. In chapter 2.2 the orientation process is explained.
The reconstruction of the model is described in chapter 3, using shape from silhouette and color image matching techniques.

2. SYSTEM CONFIGURATION

The system we use consists of a simple CCD video-camera with the ability to acquire still images. Furthermore, we need a calibration object to compute the interior orientation parameters of the camera. The object is placed in front of a homogeneous blue background to distinguish background from object pixels. The object is then rotated, resulting in a circular camera setup. In the following chapters we would like to state the camera calibration briefly, the introduction of control points on the object and finally the bundle block adjustment.

2.1 Camera Calibration

As a very basic precondition to any subsequent spatial object reconstruction, the sensor has to be calibrated. In our case, there is no way of using a calibration certificate, since we are using a standard video camera with auto-focus. Although the focus can be fixed, we cannot assure that has been unchanged since the last use.
So we calibrated it anew, using several images with a special calibration object, as shown in figure 1.

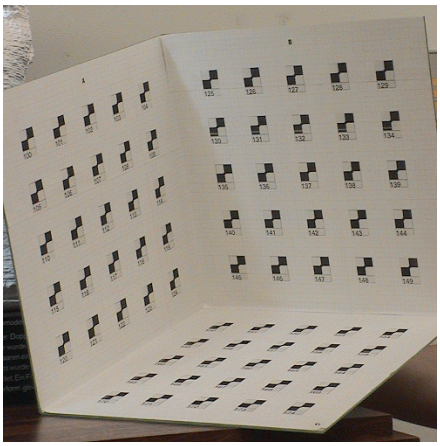

Figure 1. Calibration object with 75 spatially well distributed control points.

The camera parameters were calibrated in a bundle block adjustment with self calibration, using five images and 75 control points each. The system had 750 observations and 33 unknowns, 6 for each image and three for the camera. Additional parameters were intentionally ignored, since previous calibrations have shown that they are neglectable.
The resulting parameters for the camera were as follows:

Calibrated focal length:	$c = 34.133 \pm 0.185$ *mm*
Principal point:	$x_p = 0.211 \pm 0.146$ *mm*
	$y_p = 0.050 \pm 0.137$ *mm*

As mentioned, the focus remained fixed throughout the subsequent processes.

2.2 Image orientation

Prior to the image orientation, we had to apply some control points to the object itself. It was out of question to mark points artificially, so we had to choose 'natural' textures, instead.
We used the coordinates of the calibration object to define a local coordinate system, in which we derived the control points on the object. This is an arbitrary system, without relation to a geodetic reference system. Figure 2 illustrates the control points on the object, which served as reference for the subsequent image orientation.

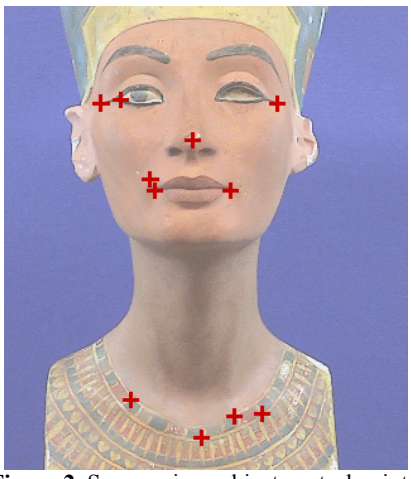

Figure 2. Some unique object control points.

For an accurate object reconstruction the exact image orientation must be known. Consequently, the images were adjusted in a bundle block adjustment, using 22 images in a circular setup. Figure 3 shows a visualization of the setup situation.

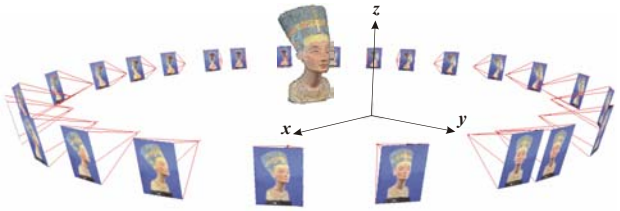

Figure 3. The virtual camera setup, using VRML-visualization.

As depicted in figure 2, we were able to use control points on the front part of the object. Using enough tie points in all the images, it was possible to perform a bundle block adjustment with all the surrounding images. With the previously calibrated camera, we managed to achieve very accurate results. The image projection centres had accuracies of 1-2 *mm*, the rotation were determined with 0.05-0.1 *gon*.

3. OBJECT RECONSTRUCTION

3.1 Image Segmentation

In our experiment, the object is rotated and the images are captured and preprocessed. First, the contour of the real object must be extracted from the input images. Therefore, a monochromatic background was used to distinguish the object from the environment. The decision if a pixel represents background or object is based upon its position in the IHS-colorspace. Since the blue background is sufficiently homogeneous, we can easily define a hue domain which is considered background. In figure 4 we show the original image on the left, and the result of the segmentation on the right.

Figure 4. Image segmentation using an IHS color space histogram. Original image (left) and the resulting silhouette extraction (right).

3.2 Shape Modelling Using Voxel Carving

When the camera geometry is known, a bounding pyramid can be constructed for every image. All voxels are projected into the every image, if the image coordinate defines a background pixel, the voxel is marked to be deleted (voting). The shape is computed volumetrically by carving away all voxels outside the projected silhouette cone (see Fig. 5). The intersection of all silhouette cones from multiple images defines estimate geometry of the object called visual hull. When the greater numbers of views are used, this technique progressively refines the object model. Finally, the voxels are purged using a threshold for the number of votes.

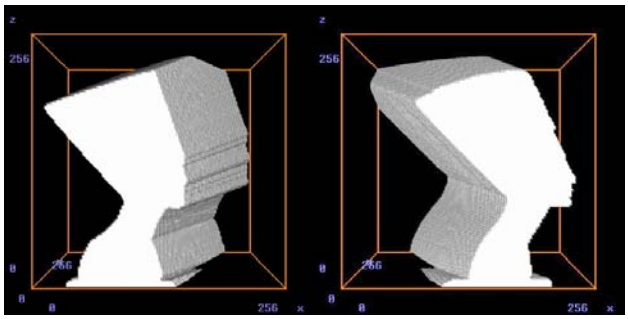

Figure 5: Voting-based carving of a voxel cube using various silhouettes under central projection.

3.3 Color Image Matching

In order to refine the model, color image matching was used to get into the visual hull and carve away the voxels in the critical areas. The image matching was done, using the normalized cross correlation, which will be explained more detailed later in this chapter. However, the search region in the second image can be narrowed down to a broad line since the image orientation is known. Although an object point corresponds to exactly one image point, the reversion is not valid. Instead, every object point along a line of sight may be projected into the same pixel. Only by the use of a second image, we are able to derive a unique point in space. But we can also use this line of sight to limit the positions in the second image, where the object point may appear. This situation is illustrated in figure 6 and it is known as the epipolar line.

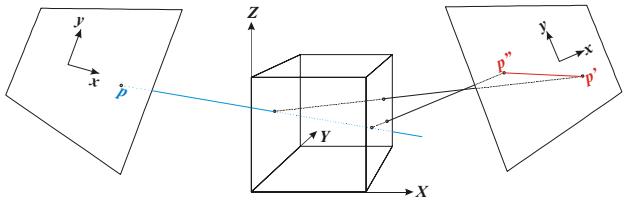

Figure 6. Epipolar geometry.

We can use this line to limit the search area for image matching, since it is a very time consuming process.
In this work we do not actually calculate the epipolar line, instead we trace the pixel of interest back into the object space step by step. For each step, we project this three dimensional coordinate into the second image, where we perform the correlation.
The following equation is used to trace pixels back into object space:

$$\lambda \begin{pmatrix} x_i - x_0 \\ y_i - y_0 \\ -c \end{pmatrix} = R \begin{pmatrix} X - X_0 \\ Y - Y_0 \\ Z - Z_0 \end{pmatrix} \tag{1a}$$

$$\Rightarrow \quad \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = R^{-1} \lambda \begin{pmatrix} x_i - x_0 \\ y_i - y_0 \\ -c \end{pmatrix} + \begin{pmatrix} X_0 \\ Y_0 \\ Z_0 \end{pmatrix}$$

From the equations above we can derive the formulas:

$$
\begin{aligned}
X &= \lambda r_{11}(x_i - x_0) + \lambda r_{12}(y_i - y_0) - \lambda r_{13} c + X_0 \\
Y &= \lambda r_{21}(x_i - x_0) + \lambda r_{22}(y_i - y_0) - \lambda r_{23} c + Y_0 \\
Z &= \lambda r_{31}(x_i - x_0) + \lambda r_{32}(y_i - y_0) - \lambda r_{33} c + Z_0
\end{aligned}
\qquad (1b)
$$

where c = calibrated focal length
X, Y, Z = object point
X_0, Y_0, Z_0 = projection centre
x_i, y_i = image point
x_0, y_0 = principal point
R, r_{ik} = rotation matrix
λ = scaling factor

Following the rules of central projection, each pixel has it's individual scale factor λ, depending on the point's distance from the projection centre. In our algorithm, we use this scale factor as a variable to loop along a line of sight into object space.
First, we define a very rough minimum and a maximum λ for each corner of the entire voxel cube as the loop limits. For means of speed optimization we start the loop with rather large steps, which will be decreased as soon as the ray of sight hits an opaque voxel. This way the loop quickly enters the bounding cube, but slows down to make sure that every foreground voxel is met. Every opaque voxel encountered on the ray of sight is stored in a list.
Each processed voxel is projected into the second image, using the below well known collinearity equations. This is the inverse transformation compared to the tracing of pixels into the cube.

$$
\begin{aligned}
x_i &= x_0 - c\frac{R_{11}(X - X_0) + R_{21}(Y - Y_0) - R_{31}(Z - Z_0)}{R_{13}(X - X_0) + R_{23}(Y - Y_0) - R_{33}(Z - Z_0)} \\
y_i &= y_0 - c\frac{R_{12}(X - X_0) + R_{22}(Y - Y_0) - R_{32}(Z - Z_0)}{R_{13}(X - X_0) + R_{23}(Y - Y_0) - R_{33}(Z - Z_0)}
\end{aligned}
\qquad (2)
$$

Only surface voxels should represent the similar regions when they are projected into the images they are visible. Consequently, non corresponding image points do not represent a surface voxel. So the decision of a voxel being on the surface of the object or not is made by the correlation coefficient, applied to a matrix around the projected image points.
A normalized cross correlation is used, since its results range from -1 to +1. Differences in brightness and contrast of the image pairs are taken into consideration. The following computer optimized equation is used to correlate the corresponding pixels.

$$
r = \frac{\sum g_1 \cdot g_2 - n \cdot \bar{g}_1 \cdot \bar{g}_2}{\sqrt{\left(\sum g_1^{\,2} - n \cdot \bar{g}_1^{\,2}\right) \cdot \left(\sum g_2^{\,2} - n \cdot \bar{g}_2^{\,2}\right)}} \qquad (3)
$$

Since the images contain full 24-bit color information, the above equation would waste information when applied to the gray value, only. Two ways have been investigated to exploit the full information. First, all three color channels have been inserted into the correlation vector, resulting in one single similarity value. The other approach calculates each channel seperately and allows individual assassment. A mean value of the three channels gives a general similarity value, like the first approach.
Correlation is based upon texture information, so in homogeneous regions a correlation coefficient will give a false result, since it always returns a high correlation factor. Consequently, it is not wise to perform cross correlation in homogeneous regions. However, in color images apparently uniform areas may vary in their seperate color channels. For example a dark blue cross on a slightly lighter blue background, will look rather uniform if this image is grayscaled, since the blue channel has a low weight in the grayscaling process. Green and red channels may be entirely empty, except for noise. But looking at the channels seperately, the blue channel will clearly reveal its texture. Experiments show that a weighted mean value of the three channels gives a reliable result. We based this weighted mean value upon the standard deviation of the gray values in the individual channels. So the color information in the search matrix is enhanced by less considering low textured channels.

When all the pixels on the epipolar line are correlated, ray tracing stops and the voxel with the maximum correlation value in the ray of sight is assigned it's corresponding color value and the previous voxels recorded in the list are marked to be deleted from voxel space. (see Fig.7)

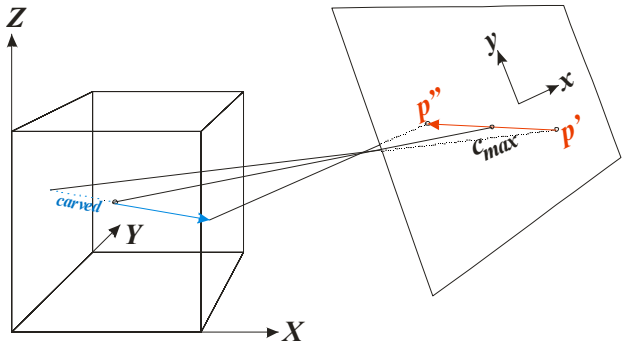

Figure 7. Carving voxel space according to image correlation.

4. RECONSTRUCTION RESULTS

Figure 8. Photorealistic reconstruction of a sculpture of Nofretete by color image matching using 22 images.

5. CONCLUSIONS AND FUTURE WORK

We presented an approach to construct photorealistic 3D models of real-world objects. The system requirements are simple therefore the method is attractive for many applications. A bundle block adjustment was performed prior to volume intersection and refined by an image matching algorithm. In order to achieve good results for the image orientations, control points and spatially well distributed tie points are measured carefully.
The calculation of the visual hull is simple and fast and gives a good approximation of the object. The ray tracing is slow, therefore further investigations should be done for speed optimization. Although the combination with the color image matching to get into the concavities gives good results for textured regions of the object, it is not reliable for non-textured regions. In the future, we would like to check the current algorithms for further automation, for example the automatic measurement of tie points and the elimination of errors in the adjustment process.

ACKNOWLEDGEMENTS

The author would like to thank O. Sinram and V. Rodehorst for their contributions, J. Moré and A. Wiedemann for their helpful assistance to the camera calibration and bundle block adjustment task.

REFERENCES

Culbertson, W. B. Malzbender, T. Slabaugh, G. , 1999. Generalized Voxel Coloring. *Proc. of the Vision Algorithms, Workshop of ICCV*, pp. 67 – 74.

Kutulakos, N. and Seitz, M. , 1998. A Theory of Shape by Space Carving. *University of Rochester CS Technical Report 692*.

Kraus, K. , 1997. *Photogrammetry. Dümmlers Verlag*, Band 2., pp. 15-19, Band 1.,pp. 349.

Kutulakos, K. N. and Seitz, S. M. , 1998. What Do N Photographs Tell Us about 3D Shape? *TR680, Computer Science Dept. U. Rochester*.

Kuzu, Y. Rodehorst, V. , 2001. Volumetric Modeling using Shape from Silhouette. *Fourth Turkish-German Joint Geodetic Days*., pp. 469-476.

Kuzu, Y. Sinram, O. , 2002. Photorealistic Object Reconstruction using Voxel Coloring and Adjusted Image Orientations. *ACSM/ASPRS Annual Conference*, Washington DC, Proceedings CD-ROM, Proceed\00437.pdf.

Laurentini, A. , 1995. How far 3D Shapes Can Be Understood from 2D Silhouettes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.17, No.2.

Matusik, W. Buehler, C. Raskar, R. Gortler, S. J. and McMillan, L. , 2000. Image-Based Visual Hulls. *SIGGRAPH 2000 , Computer Graphics Proceedings, Annual Conference Series*, pp. 369-374.

Niem, W. , 1994. Robust and Fast Modeling of 3D Natural Objects from Multiple Views. *Proceedings "Image and Video Processing II"*, Vol. 2182, pp. 388-397.

Seitz, M. Dyer, R. , 1997. Photorealistic Scene Reconstruction by Voxel Coloring. *Proceeding of Computer Vision and Pattern Recognition Conference*, pp. 1067-1073.

Slabaugh, G. Culbertson, B. Malzbender, T. and Schafer, R. , 2001. A survey of methods for volumetric scene reconstruction from photographs. In *International Workshop on Volume Graphics*, Stony Brook, New York.

Szeliski, R. , 1997. From images to models (and beyond): a personal retrospective. *Vision Interface '97, Kelowna, British Columbia, Canadian Image Processing and Pattern Recognition Society*, pp. 126-137.

Szeliski, R. , 1991. Shape from rotation. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'91),* pp. 625-630.

Vedula, S. Rander, P. Saito, H. and Kanade, T. , 1998. Modeling, Combining, and Rendering Dynamic Real-World Events From Image Sequences. *Proc. 4th Conference on Virtual Systems and Multimedia (VSMM98)*, pp. 326-332.