

High-quality Structured-light Scanning of 3D Objects using Turntable

Csaba Kazó

Geometric Modelling and Computer Vision Laboratory
MTA SZTAKI
Budapest, Hungary
Email: kazó@vision.sztaki.hu

Levente Hajder

Geometric Modelling and Computer Vision Laboratory
MTA SZTAKI
Budapest, Hungary
Email: hajder.levente@sztaki.mta.hu

Abstract—This study presents a 3D structured light scanner that can reconstruct 3D objects. The goal of the paper is to show that high quality 3D models can be computed using the equipment proposed here. The resulting 3D object then can be used in arbitrary 3D applications such as augmented reality or human-computer interaction solutions.

The object to be reconstructed is placed on a turntable. It is illuminated by a standard projector and images are taken of it by a high resolution color digital camera. The turntable rotates the object, enabling measurements from different points of view. The proposed scanning system is fully automatic, no manual interaction is required. A novel rendering method is also presented that can generate realistic images of the reconstructed model from its colored point set representation.

I. INTRODUCTION

The area of cognitive infocommunications [1] is the link between infocommunications and cognitive sciences. A very important subproblem of the cognitive computer sciences is augmented reality, when parts of the real and synthetic world are presented to a human at the same time. The quality of the augmented reality applications significantly depends on the visualized synthetic models. A high quality model increases the efficiency and comfort of a user while interacting with the computer. It is a key problem in many applications, such as 3D Internet [2].

This paper deals with the problem of reconstructing real-world spatial object. Namely, it describes the SZTAKI 3D scanner and shows that it can reconstruct the colored three dimensional model of static lambertian objects. The apparatus contains a turntable that holds and rotates the object, and a commercial projector used to illuminate it. The advantage of our system is that 360-degree reconstruction is obtained thanks to the rotating table.

It is not the first equipment for turntable-based reconstruction, but existing solutions [3], [4] do not involve any illumination, they only utilize a digital camera. The novelty of the SZTAKI 3D scanner is that the illuminated patterns make the reconstructed objects more precise.

II. PROBLEM STATEMENT

The aim of the scanner is to build very accurate colored 3D models from images taken of spatial object. The final 3D model is represented by a colored point set P_i , $i = 1 \dots P$

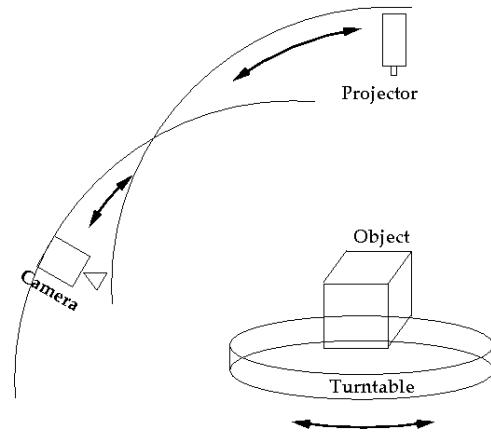


Fig. 1: Architecture of the SZTAKI 3D scanner

where P is the number of points. Each point P_i is represented by 3D coordinates $[X_i, Y_i, Z_i]^T$ and a color described with three components $[R_i, G_i, B_i]^T$.

III. OVERVIEW OF THE EQUIPMENT

The main structure of the 3D scanner is visualized in Fig. 1. The apparatus contains two moving arms and a turntable. The object to be reconstructed is placed on the turntable, the camera is held by the first arm, while the projector is fixed on the second arm.

The projector illuminates the object with patterns. There are different types of patterns such as stripes [5] or blobs [6]. We use stripes in our approach due to its simplicity. (However, we will propose a blob-based solution in the future since unstructured patterns can be used to reconstruct specular objects. On the other hand, when using blobs, the time demand of reconstruction is significantly higher.)

The camera and the projector are fixed on their respective arms, but these arms can rotate in order to illuminate and examine the object from different directions.

IV. CALIBRATION OF THE COMPONENTS

Three components need to be calibrated in the current state of our 3D scanner: the camera, the projector, and the turntable. In order to calibrate the whole system, a reference coordinate system must be selected. For practical reasons, we chose the

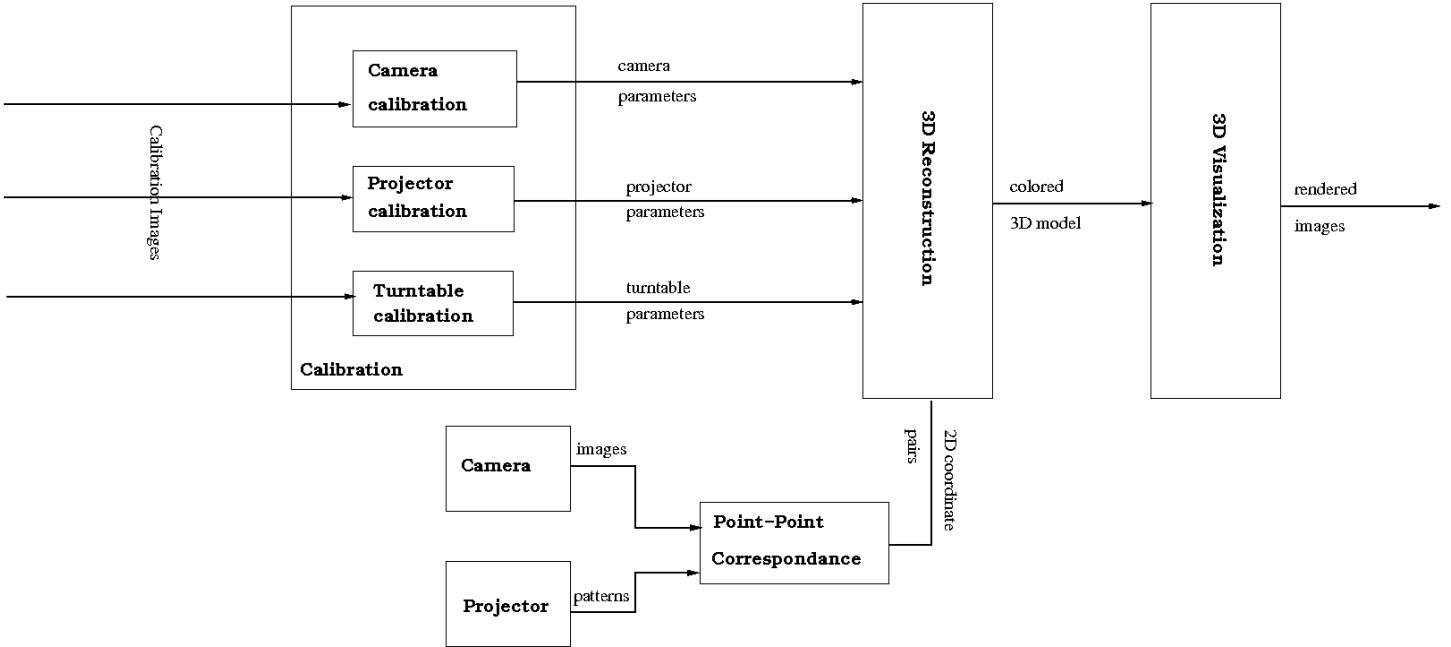


Fig. 2: Components of the reconstruction method

the camera at the zero position of its arm: the focal point is the origin, and the two axes of the camera plane give the orientation of the first and second 3D axes. The third axis of the coordinate system is aligned with the viewing direction of the camera.

A. Calibration of the Camera

The well-known Zhang calibration method [7] is used to compute the intrinsic and extrinsic parameters of the 5Mpixel digital camera of the SZTAKI scanner. The calibration procedure assumes that a chessboard pattern is captured by the camera at different positions and orientations as it is pictured in Fig. 3. The corners of the chessboard are detected using the open source OpenCV library [8]. The projection equation between the 3D coordinates of the points and the homogeneous 2D projective coordinates can be written as follows:

$$\begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} \sim C[R|t] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (1)$$

where the \sim sign denotes the equality up to scale.

Our system handles the so-called radial parameters of the camera optics as well. In this extended model, the image coordinates $[u_i v_i]$ can be written as the nonlinear function of the projective coordinates

$$\begin{aligned} u &= u'(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + 2p_1 u'v' + p_2(r^2 + u'^2) \\ v &= v'(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + 2p_2 u'v' + p_1(r^2 + v'^2) \end{aligned}$$

where $r^2 = u'^2 + v'^2$. (The nonlinear distortion of the camera is often called radial distortion because it depends on the “radius” r , the distance from the optical axis.)

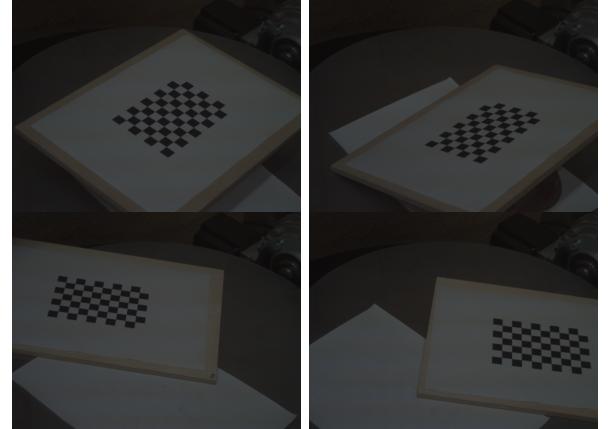


Fig. 3: Chessboard pattern for camera calibration

The calibration requires at least three images of the chessboard pattern. It first solves a linear problem to obtain initial estimates of the camera parameters, and then it applies the well-known Levenberg-Marquardt numerical optimization algorithm to find the accurate solution.

Note that the Zhang method [7] assumes that the chessboard pattern is fixed in space and the camera is moving. It is not true in our case: the camera is in fact stationary on the corresponding arm. However, this is not a problem: by using the camera positions and orientations provided by the calibration, all the results can be transformed to a common coordinate system relative to an arbitrarily placed camera.

B. Calibration of the Projector

The basic idea for projector calibration is that the projector is an inverse camera: the camera projects the 3D world into

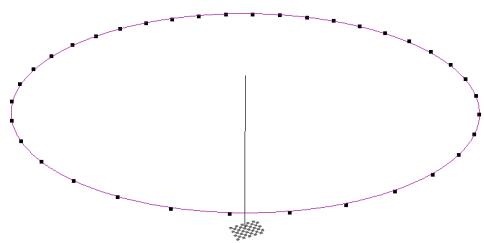


Fig. 4: Estimated camera positions and the rotation axis

a 2D plane, while the projector casts its 2D input image into the 3D world. The projection equations of the two instruments are the same. Therefore, the calibration of the projector also requires correspondences between the “image plane” of the projector (the image that is projected) and the 3D world. Pixels in the projector’s image plane are identified in the photographs by using regular stripe patterns that assign unique intensity sequences to each projector pixel, and projecting these patterns onto the known plane of the chessboard. The 3D positions of the projected pixels can be calculated from their position in the camera image and the parameters of the camera. These correspondences are the input of the same calibration algorithm that was used to obtain the camera parameters from the chessboard corners.

C. Calibration of the Turntable

The goal of the turntable calibration is to compute the axis of rotation. For this purpose, the calibration chessboard is placed on the turntable, it is rotated around 360 degrees in uniform increments, and photographs of it are added to the camera calibration image sequence.

As turntable rotates the chessboard pattern, the camera positions form the points of a circle in the coordinate system fixed to the chessboard. Since the plane of the turntable coincides with the chessboard plane in these images, the calibration method only needs to find the center of this circle; the axis is the line that traverses this center point and is perpendicular to the plane of the turntable. There are several methods for finding the circle from a collection of its points [9], [10]. However, since we performed the rotation in such a way that the points are placed uniformly on the circumference, in our case the center is simply the centroid of these camera positions.

V. POINT-POINT CORRESPONDANCES

Finding the correspondance between the camera and the projector pixels is necessary both for the calibration of the projector and the reconstruction of objects. See [5] for an overview of the possible solutions to this problem. We selected the stripe-based approach, in which the projector illuminates vertical and horizontal stripes of varying widths. The exact patterns are selected such that each pixel is encoded with a unique sequence of bits. The code is obtained by concatenating the binary representation of the pixel’s column index and row

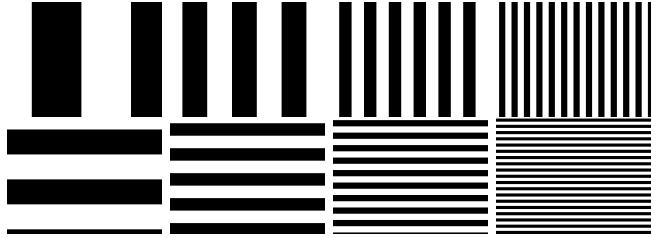


Fig. 5: Projected stripes

index. Then the stripes are assembled such that the color of a pixel in the i-th projector pattern is black if the i-th bit in the code of a pixel is 0, and white if it is 1. See Fig 5 for some examples.

To find the point-point correspondances, the task is to determine the projected code of each pixel in the photographs, and this in turn requires differentiating whether a particular point is illuminated by the projector or not. First, the area that is visible from the projector is established by using a completely black and a completely white projection. The scene is captured by the camera, and all pixels in the photograph are evaluated: if the difference between the intensity of a pixel in the dark and the bright image is smaller than a threshold, then it is assumed that the corresponding 3D point is either in shadow or outside the projection area, and so it is ignored in future calculations.

The second step of the pairing algorithm is to project the stripe patterns onto the object. The corresponding bit of the pixels is given by whether the pixel is lit or unlit, and this is determined by comparing the observed intensity of the point with its intensities measured with the completely black and completely white projections.

The above method is repeated for each stripe pattern, and results in the complete code of the pixels, from which their coordinates can easily be decoded. This method is used to compute the projector pixels corresponding to every visible point.

VI. 3D RECONSTRUCTION BY TRIANGULATION

The method described in the previous section gives us the projection of the unknown 3D points in both the camera’s and the projector’s image plane. From the point of view of the camera, we obtain a 3D line which we know contains the points that can possibly be projected onto that particular pixel. Similarly, from the point of view of the projector, the line corresponding to the projector pixel specifies the direction in which it is projected. The 3D point has to lie on both of these lines. Since due to inaccuracies these lines do not intersect, the point which is closest to them both (in the sense of minimal sum of square distances) is accepted as the result of the reconstruction.

VII. VISUALIZATION OF THE RECONSTRUCTED OBJECT

When displaying the result of the reconstruction, the first step is to render the point cloud using OpenGL. This produces an image such as the one in the left of Fig 6. As a second step,



Fig. 6: Original colored 3D model (left) and improved one (right)

this image is processed by our algorithm which attempts to extrapolate the image data to fill in holes. Each pixel is processed in the same fashion, which makes the algorithm suitable for pixel-wise concurrent execution on graphics hardware.

If the pixel belongs to the background, then the program searches its neighborhood to find the closest non-background pixel, and at the same time to counts the non-background pixels in every direction relative to the current pixel. If at least one of the directions has too few points, that suggests that we are at the edge or outside the object, and no operation is performed. Otherwise the color of nearest pixel is applied.

This approach fails to handle the case where there are separate surfaces behind one another. Then holes are filled from both from the front and the back point colors, mixing two unrelated textures together. To remedy this, we utilize the depth information (which is also available from the OpenGL render result) and extend the program to only consider points which are in the front. Before the search described in the previous paragraph, another search is performed which computes the range of depth values around the current pixel. The median depth is then used to ignore points which are in the back. By performing this correction even for non-background pixels, we can fill holes through which not the background but a different surface is visible. Figure 6 shows how the original colored 3D point cloud can be improved by the proposed visualization algorithm.

VIII. RESULTS

The testing machine was a standard PC with 2.33GHz Intel Core 2 Quad CPU and NVIDIA GeForce GTX 285 GPU. The maximal operating frequency of the GPU is 1.48 GHz, and it consists of 240 CUDA cores. We used an image size of 800×800 to render the results. For this size, the visualization algorithm requires at most 150ms per frame, which means that the final result can be presented at 5 – 6 frames per second in the worst case.

We did not consider the time demand of the 3D reconstruction itself critical, and used a straightforward single-threaded implementation which takes a few minutes to compute models in the range of hundreds of thousands to millions of points.

We tested the quality of our scanner on two real-world objects. The first one is a hand-painted cube shown in Fig. 7. The length of its edges is approximately 20 centimeters. 4 of



Fig. 7: Object 'Cube' illuminated by stripes



Fig. 8: Rendered images of 3D colored model of 'Cube'

the input images are presented in Fig. 7. Fig. 8 illustrates the reconstructed 3D model.

The second test object is a flacon (see Fig. 9). Its height is approximately 30 centimeters. The 3D model is pictured in Fig. 10.

There are several factors that hinder this kind of reconstruction. Some areas of the surface are always in shadow from the projector, and some are not in view of the camera despite being illuminated because of the distance between the two devices. Furthermore, very dark textures cannot be illuminated strongly enough to make lit and unlit pixels distinguishable. Despite the fact that all of these issues prevent reliable reconstruction of the corresponding surface areas, the use of the turntable and the visualization algorithm helped achieve a mostly complete result.

IX. CONCLUSION AND FUTURE WORK

We have shown that the SZTAKI scanner can produce high quality and realistic 3D models of real-world objects. This



Fig. 9: Object 'Flacon' illuminated by stripes



Fig. 10: Rendered images of 3D colored model of 'Flacon'

study has described the mechanical and algorithmical structure of the scanner as well as the operation of its components. The quality of the reconstruction has been demonstrated on two real examples.

In the future, we will propose a calibration method for the arms in order to be able to move the camera and the projector during the reconstruction process so that even more data can be gathered. Furthermore, we plan to deal with surfaces with non-lambertian effects such as specularity and transparency.

ACKNOWLEDGMENT

The authors would like to thank prof. Dmitry Chetverikov for his useful council.

This work was supported by the NKTH-OTKA grant CK 78409.

REFERENCES

- [1] P. Baranyi and A. Csapo, "Cognitive infocommunications: Coginfocom," in *11th International Symposium of Hungarian Researchers on Computational Intelligence and Informatics*, 2010.
- [2] P. Baranyi, B. Solvang, H. Hashimoto, and P. Korondi, "3d internet for cognitive infocommunication," in *10th International Symposium of Hungarian Researchers on Computational Intelligence and Informatics*, 2009, pp. 229–243.
- [3] A. W. Fitzgibbon, G. Cross, and A. Zisserman, "Automatic 3D model construction for turn-table sequences," in *3D Structure from Multiple Images of Large-Scale Environments*, 1998, pp. 155–170.
- [4] V. Fremont and R. Chellali, "Turntable-based 3d object reconstruction," in *IEEE Conference on Cybernetics and Intelligent Systems*, 2004, pp. 1277–1282.
- [5] S. Zhang, "High Resolution, Real-Time 3D Shape Measurement," Ph.D. dissertation, Stony Brook University, 2005.
- [6] V. Chapdelaine-Couture, N. Martin, and S. Roy, "Unstructured light scanning to overcome interreflections," in *International Conference on Computer Vision*, 2011, pp. 1895–1902.
- [7] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. on PAMI*, vol. 22, no. 11, pp. 1330–1334, 2000.
- [8] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.
- [9] A. Fitzgibbon, M. Pilu, and R. Fisher, "Direct Least Square Fitting of Ellipses," *IEEE Trans. on PAMI*, vol. 21, no. 5, pp. 476–480, 1999.
- [10] N. Chernov and C. Lesort, "Least squares fitting of circles," *Journal of Mathematical Imaging and Vision*, vol. 23, no. 3, pp. 239–252, 2005.