Modelling and Calibration of Multi-Camera-Systems for 3D Industrial Supervision Applications

Guido Straube[a], Chen Zhang[b], Artem Yaroshchuk[a], Steffen Lübbecke[a], and Gunther Notni[b]

[a]SQB GmbH, Werner-von-Siemens-Str. 9, 98693 Ilmenau, Germany
[b]TU Ilmenau, Fakultät für Maschinenbau, Fachgebiet Qualitätssicherung und Industrielle Bildverarbeitung, Gustav-Kirchhoff-Platz 2, 98693 Ilmenau, Germany

ABSTRACT

With the advent of industry 4.0, the introduction of smart manufacturing and integrated production systems, the interest in 3D image-based supervision methods is growing. The aim of this work is to develop a scalable multi-camera-system suitable for the acquisition of a dense point cloud representing the interior volume of a production machine for general supervision tasks as well as for navigation purposes without a priori information regarding the composition of processing stations. Therefore, multiple low-cost industrial cameras are mounted on the machine housing observing the interior volume. In order to obtain a dense point cloud, this paper reviews aspects of metric stereo calibration and 3D reconstruction with attention being focused on target-based calibration methods and block matching algorithms.

Keywords: Camera Calibration, Stereo Vision, Computer Vision, Dense Matching, Scene Reconstruction, Point Cloud, OpenCV, Python

1. INTRODUCTION

As low-end image sensors are available in many different configurations, the approach of using multiple cameras to increase the depth of information captured by the image sensors becomes more common. The Stanford University led this approach with the publication [1] presenting a dense array of CMOS image sensors used to capture high-speed videos and later with [2] using 100 cameras for synthetic aperture imaging. Besides these camera arrays utilizing the increased depth of information from more sensors, binocular stereo computer vision systems are common. Principally, two views are enough to compute 3D information of an object depicted in both frames. Therefore a rig of two cameras can be used or the camera has to move around the object to capture frames from different viewpoints. The latter principle is commonly referred to as structure from motion, whereas the primal approach is called stereoscopy. As both principles rely on at least two views with a common area, software implementations for structure from motion can, at least in parts, be utilized for stereoscopy and vice versa as they share the same mathematical approach.

In the presented work the open source computer vision library OpenCV in version 3.4.4 is used. The multi-camera system referred to in this paper is designed to overview the interior of a production

Further author information: (Send correspondence to G.S.)
G.S.: E-mail: guido.straube@quick-image.de, Telephone: +49 3677 469 059-18
C.Z.: E-mail: chen.zhang@tu-ilmenau.de, Telephone: +49 3677 69 39-77
A.Y.: E-mail: artem.yaroshchuk@tu-ilmenau.de
S.L.: E-mail: steffen.luebbecke@quick-image.de, Telephone: +49 3677 46 905-90
G.N.: E-mail: gunther.notni@tu-ilmenau.de, Telephone: +49 3677 69 38-20

Photonics and Education in Measurement Science 2019, edited by Maik Rosenberger, Paul-Gerald Dittrich, Bernhard Zagar, Proc. of SPIE Vol. 11144, 111440D
 · doi: 10.1117/12.2532320

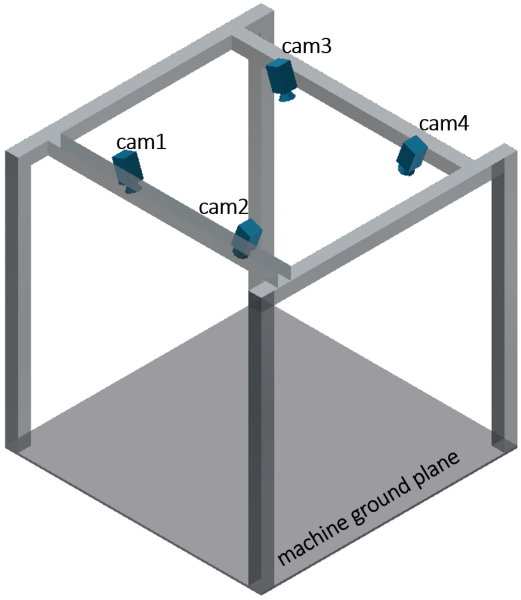

Figure 1. The conceptual layout of the multi-camera-system with four cameras mounted to the machine frame is depicted. From these four individual cameras six different combinations of camera pairs evolve: 1-2; 1-3; 1-4; 2-3; 2-4; 3-4

machine with tools moving on a gantry in X, Y, Z direction. Multiple processing stations may be placed in the range of motion of the gantry. Since the machine should be able to work its way through the interior without colliding with any object, the three-dimensional profile needs to be detected. Therefore multiple cameras, at first four, are mounted to the machine frame (see Figure 1). A stereovision approach is used, which leads to six stereo pairs for four cameras.

Generally speaking, the aim of stereoscopy algorithms is to calculate a disparity map from two frames by two cameras corresponding to the same scene. The disparity map holds information about the different positions of common scene points projected to the individual camera frames. With this information the point can be easily reprojected to three-dimensional coordinates, resulting in a point cloud with every point containing X, Y, Z coordinates in real world coordinates. In the presented setup this leads to six point clouds from six stereo pairs depicting the machine interior from different viewpoints.

2. CAMERA CALIBRATION

The calibration of a multi-camera system is an essential part in the process of reconstructing three-dimensional data from two-dimensional images, as it defines the possible measurement accuracy as well as the scale. The aim of the calibration is to get the intrinsic and extrinsic parameters of every camera to transform a point from world coordinates to image pixel coordinates (see Figure 2). Later these parameters are used to undistort and rectify the images, along with for image transformations necessary for the reconstruction of 3D points.

The extrinsic parameters include the rotation matrix $\boldsymbol{R}$ and translation vector $\boldsymbol{t}$. Intrinsic parameters are the camera matrix $\boldsymbol{A}$, comprising the focal lengths f_x, f_y and the optical center (c_x, c_y), as well as the distortion coefficients $\boldsymbol{d}$.
All calibration parameters stay true as long as the position and orientation of the cameras, along with

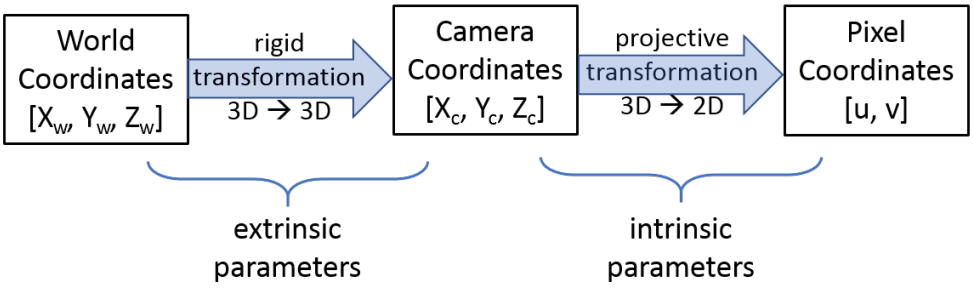

Figure 2. Overview of coordinate transformations for camera calibration.

the focal settings, do not change. Therefore the calibration has to be carried out only once for a camera setup.

2.1 Single Camera Calibration

To start the camera calibration, a camera model has to be chosen and be described. The OpenCV calibration algorithm utilizes a pinhole camera model and introduces radial and tangential distortion [3]. Distortion correction is vital since the presented multi-camera system uses low-cost board-level cameras with S-mount lenses, which introduce an amount of distortion to the images that can not be neglected.

At first a transformation from a three-dimensional point $P_w(X_w, Y_w, Z_w)$ in world coordinates to a point $P(u, v)$ in image pixel plane coordinate system has to be found (see Figure 3). The equation

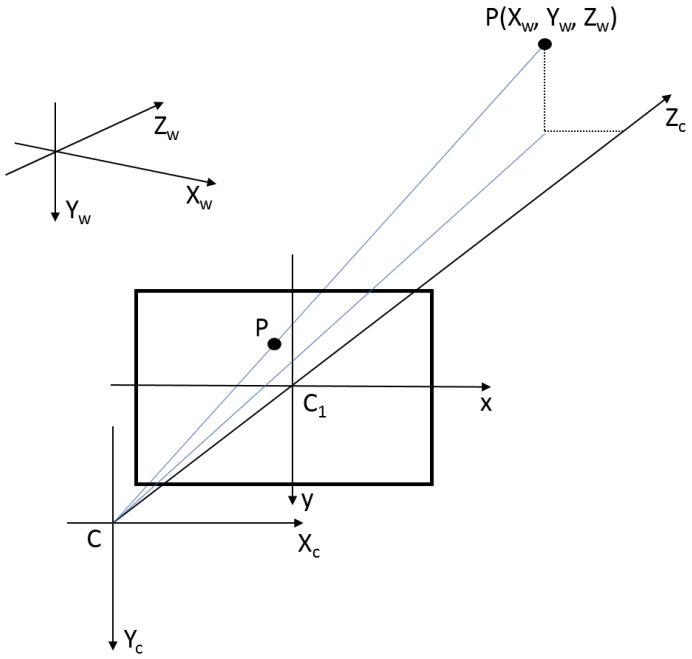

Figure 3. Coordinate system for camera calibration

(1) transforms a point P_w to a point $P_c(X_c, Y_c, Z_c)$ in the camera coordinate system, where R is a $3x3$ rotation matrix and t is a $3x1$ translation vector.

$$\begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} = R \begin{pmatrix} X_w \\ Y_w \\ Z_w \end{pmatrix} + t \tag{1}$$

The point P_c is now projected through the pinhole model in order to obtain physical coordinates on the image plane as $P(x, y)$, see (2).

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} X_c/Z_c \\ Y_c/Z_c \end{pmatrix} \tag{2}$$

By introducing radial distortion coefficients (k_1, k_2, k_3), the corrected point coordinates $P_k(x_k, y_k)$ are defined as following:

$$\begin{aligned} x_k &= x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \\ y_k &= y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \end{aligned} \tag{3}$$

With k_1, k_2 and k_3 being the radial distortion coefficients and $r^2 = x^2 + y^2$. Due to a lens not aligned perfectly to the image plane, tangential distortion is introduced. Its correction for a point $P_p(x_p, y_p)$ can be described as:

$$\begin{aligned} x_p &= x + (2p_1 xy + p_2(r^2 + 2x^2)) \\ y_p &= y + (p_1(r^2 + 2y^2) + 2p_2 xy) \end{aligned} \tag{4}$$

In summary the distortion coefficients are defined as $d = (k_1, k_2, p_1, p_2, k_3)$. The tangential and radial distortion corrected point $P_q(x_q, y_q)$ is, as combination of the equations (3) and (4), defined as:

$$\begin{pmatrix} x_q \\ y_q \end{pmatrix} = (1 + k_1 r^2 + k_2 r^4) \times \begin{pmatrix} X_c/Z_c \\ Y_c/Z_c \end{pmatrix} + \begin{pmatrix} 2p_1 xy + p_2(r^2 + 2x^2) \\ p_1(r^2 + 2y^2) + 2p_2 xy \end{pmatrix} \tag{5}$$

To translate the physical image plane coordinates to image plane pixel coordinates, the camera matrix A is needed, which contains information about the focal length in mm as f_x and f_y, as well as the optical center in pixel coordinates as (c_x, c_y). γ represents the skewness, which is the angle error in between the two axis of the pixel array. For industrial grade image sensors skewness is usually small and can be neglected.

$$A = \begin{bmatrix} f_x & \gamma & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

In conclusion a point $P(u, v)$ in pixel coordinates is defined as:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} f_x x_q + c_x \\ f_y y_q + c_y \end{pmatrix} \tag{7}$$

2.2 Stereo Camera Calibration

Stereo camera calibration, or binocular calibration, will, in addition to the already obtained intrinsic and extrinsic parameters, get the relative position of one camera to the other of a stereo pair along with matrices necessary for the reconstruction of the scene.

We suppose the left camera extrinsic parameters are rotation matrix R_L and translation vector t_L. For the right camera these are R_R and t_r. From these matrices and vectors the translation vector from left to right camera coordinates $\boldsymbol{t_{LR}}$ and the corresponding rotation matrix $\boldsymbol{R_{LR}}$ are derived.

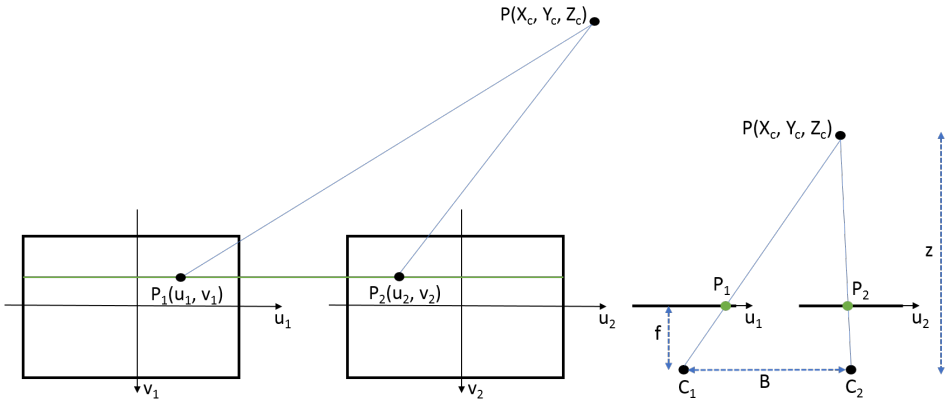

Figure 4. Left side: point P projected onto rectified frames of stereo pair; right side: point P projected onto rectified frames of stereo pair in top view.

The keyword for stereoscopic reconstruction is epipolar geometry, which encapsulates the relation of the projective geometry between two views. The centrepiece of this is the fundamental matrix $\boldsymbol{F}$, which derives from the relation given in equation (8), with x as the projection of a real-world point X in the left camera and x' in the right camera frame. The equation could be solved for image points with known real world distances obtained from both cameras of a stereo pair. For a more in-depth explanation of epipolar geometry refer to [4].

$$x'^T F x = 0 \tag{8}$$

To be able to transform images pairwise abiding to epipolar geometry, rectification homographies $\boldsymbol{H_L}$ and $\boldsymbol{H_R}$ need to be found. With known fundamental matrix F, the algorithm described in [5] can be utilized.
Figure 6 shows frames of pair 2 distortion corrected and rectified, which means the rectification homographies are applied.
To be able to reproject image points to 3D coordinate space, disparity has to be introduced. Figure 4 depicts a simplified projection of a point P onto two already rectified images. Since these images are rectified, the projections P_1 and P_2 are located on one epipolar line. Therefore $v_1 = v_2$ and $u_1 \neq u_2$. The equation for disparity is given by (9) with B as baseline or distance between the optical centers of the cameras (C_1 and C_2), f as focal length and z as distance to point P in 3D world coordinate system.

$$u_1 - u_2 = \frac{B \cdot f}{z} = disparity \tag{9}$$

One step in stereo calibration is still missing, the computation of the disparity-to-depth mapping matrix $\boldsymbol{Q}$. Its definition is given by equation (10). Since the vectors are noted in homogeneous coordinates their fourth entry refers to scale. With this equation a disparity map can be reprojected to the world coordinate system in 3D (see section 3).

$$\begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} = Q \cdot \begin{bmatrix} u \\ v \\ disparity(u,v) \\ 1 \end{bmatrix} \tag{10}$$

2.3 Calibration Procedure

In order to apply the above elucidated relations, a set of points with known distances is needed. Therefore a chessboard type calibration target is used, as shown in Figure 5. A set of images with different positions of the target are taken, where the target should be moved to every position in the measurement volume in order to get a strongly calibrated stereo vision system. The position of the corners on the calibration target are detected by using the OpenCV function *findChessboardCorners()* and, to get the corner position with subpixel precision, *cornerSubPix()* is called afterwards. This routine is carried out for every image during the calibration process. This leads to a set of points for every camera where every point refers to a corner on the chessboard pattern. With these points and an array with distance values in *mm* corresponding to the dimension of the rectangles in the calibration pattern, the cameras can be calibrated. For every stereo pair at first the extrinsic and intrinsic parameters of the two partaking cameras are calculated (see section 2.1) using the OpenCV function *calibrateCamera()*, which is based on [6] and [7]. This routine delivers the camera matrices A_L and A_R, distortion coefficients d_L and d_R, rotation matrix and translation vector for every pattern view and a reprojection error. The distortion coefficients are later used to undistort images, as depicted in Figure 6 on the left side.
From the rotation and translation matrices for every pattern view, the rotation matrix R_{LR} and translation vector t_{LR} from left to right camera are derived.

The above mentioned chessboard corner points in image pixel coordinates are now undistorted using the function *undistortPoints()*, and further on used to obtain the F matrix via the OpenCV function *findFundamentalMat()*. This function utilizes the random sample consensus (RANSAC) algorithm to solve the problem delineated in equation (8). With the obtained F matrix the above mentioned rectification homographies can be calculated. In OpenCV the function *stereoRectifyUncalibrated()* computes the rectification homographies H_L and H_R using the algorithm presented in [5]. On the right side Figure 6 depicts rectified images of stereo pair 2.

After rectification a disparity map can be calculated using a matching algorithm, for example semi global block matching (SGBM), this procedure is explained in section 3. As mentioned above, the Q matrix is needed to reproject a disparity map to 3D. The connection between disparity, world points and Q is given in (10). Since we are using the OpenCV function *stereoRectifyUncalibrated()*, which does not return a Q matrix, the disparity-to-depth mapping matrix needs to be calculated in another way. Therefore the equation (10) is utilized. At first the points obtained from the calibration target are triangulated using OpenCVs *triangulatePoints()*, which results in a matrix with points in homogeneous coordinates, notated as B in (11). The matrix D in the same equation is acquired by applying H_L,

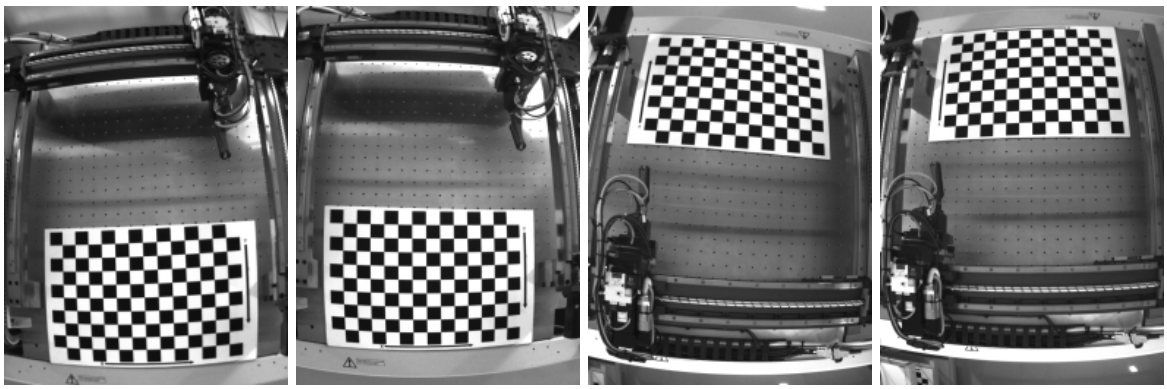

Figure 5. Frames of camera 1 to 4 (left to right) with chessboard calibration pattern. The image frames are uncorrected, neither undistortion nor rectification transformations are applied.

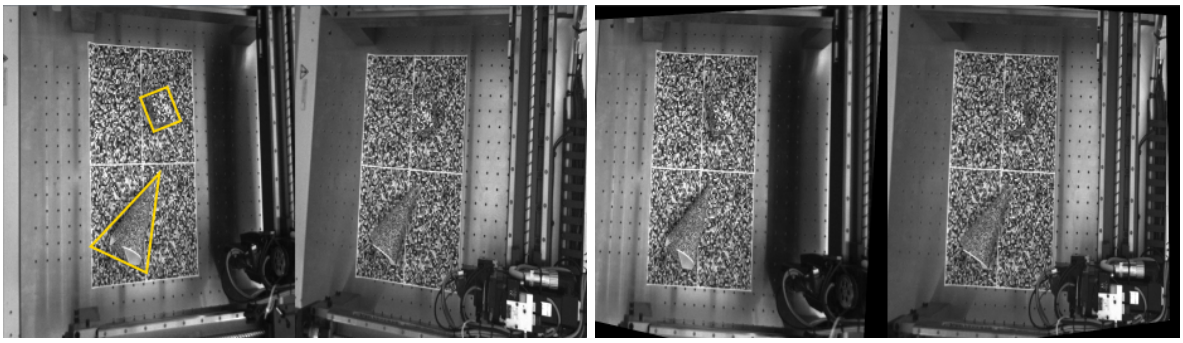

Figure 6. Images obtained from stereo pair 2 (left: camera 1, right: camera 3). The image pair shown on the left side is distortion correct, the pair on the right side is rectified. On the greyscale pattern two objects (one cube and one cone) are placed, see yellow markers in left image.

respectively H_R, to image points obtained by *findChessboardCorners()* and *cornersSubPix()*. Equation (12) describes the derivation of a pseudoinverse matrix, which is in (13) used to convert equation (11) to obtain Q. In this way it is possible to calculate the disparity-to-depth mapping matrix based on the points retrieved from the different views of the calibration target. Assuming the calibration target was moved through the whole observed volume, maximum and minimum disparity values can also be calculated using these points.

$$B = Q \cdot D$$

$$\begin{bmatrix} X_1 & X_2 & ... & X_n \\ Y_1 & Y_2 & ... & Y_n \\ Z_1 & Z_2 & ... & Z_n \\ W_1 & W_2 & ... & W_n \end{bmatrix} = Q \cdot \begin{bmatrix} u_1 & u_2 & ... & u_n \\ v_1 & v_2 & ... & v_n \\ disparity(u_1, v_1) & disparity(u_2, v_2) & ... & disparity(u_n, v_n) \\ 1 & 1 & ... & 1 \end{bmatrix} \tag{11}$$

$$D^+ = D^T \cdot (D \cdot D^T)^{-1} \tag{12}$$

$$\begin{aligned} B \cdot D^+ &= Q \cdot D \cdot D^+ \\ B \cdot D^+ &= Q \end{aligned} \tag{13}$$

3. RECONSTRUCTION OF THE SCENE

The calibration as a whole delivers information to rectify camera frames according to epipolar geometry. From this point a disparity map for a stereo pair needs to be calculated, which then can be reprojected to 3D world coordinates with the already obtained disparity-to-depth mapping matrix Q.
The section 3.1 exemplifies the computation of the disparity map, section 3.2 explains the reprojection to 3D world coordinates.

3.1 Constructing Disparity

The OpenCV implementation of the semi-global block matching (SGBM) algorithm is applied to the rectified images of stereo pair 2 (see Figure 6, right side). The OpenCV class *StereoSGBM* uses a modified algorithm from [8], whereas, instead of mutual information cost function, a simpler sub-pixel metric from [9] is implemented.

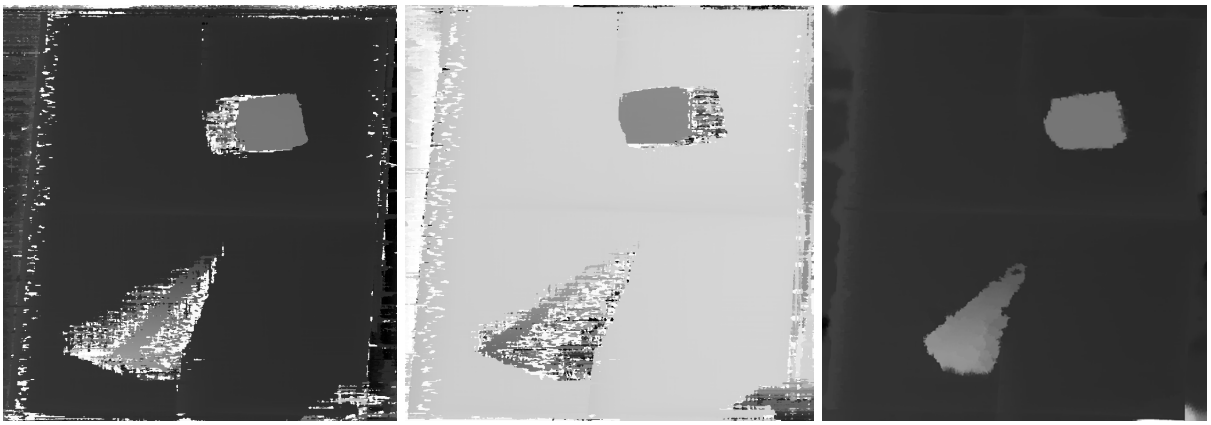

Figure 7. Disparity map of pair 2, corresponding to image pairs shown in Figure 6. From left to right these images are left-right disparity map, right-left disparity map and the filtered disparity map. The disparity is mapped to values ranging from 0 to 255, hence dark pixel represent areas further away from the camera pair, brighter pixel represent areas closer to the camera pair (except for the right-left matched map). In order for the SGBM algorithm to work properly, the surface has to be structured or a structure has to be projected. In these frames this is not implemented, hence the only valid section is the structured area (see Figure 6).

In the first step pixelwise cost calculation is done, the cost is calculated as the absolute minimum difference of intensities in the range of a half pixel in 5 directions along the epipolar line. As the name of the algorithm suggests, a global smoothness constraint is approximated. The smoothed cost for a pixel (or block, depending on the chosen block size) and disparity is calculated by summing the costs of all minimum cost paths that end in the pixel or block. The disparity map is determined by selecting a disparity with corresponding minimum cost for every pixel of the source image. This process is done two times, the first time from left to right image, the second time from right to left image. Therefore a left and right matcher instance is created. The input minimum disparity value is obtained from the matrix D in equation (11), as well as the number of disparities, which is the maximum disparity value minus the minimum disparity value. From these two maps one combined disparity map is preserved using the Weighted Least Squares filter with a left-right-consistency-based confidence map. In OpenCV these functions are implemented in the *ximgproc.DisparityWLSFilter* class. Figure 7 depicts the output disparity maps.

3.2 Constructing Point Clouds

The obtained disparity images (see Figure 7) need to be reprojected to 3D world coordinates. Therefore the disparity-to-depth mapping matrix Q is calculated, see section 2.2. In OpenCV the function *reprojectImageTo3D()* transforms a disparity map to a 3-channel point cloud. For each pixel (u, v) and its corresponding disparity value $disparity(u, v)$ the equation (10) is solved. This results in a dense point cloud with 5.038.848 points. Figure 8 depicts the valid part of the resulting point cloud.

4. CONCLUSION AND FUTURE WORK

This paper has presented a calibration technique for stereoscopic applications, utilizing the functions and algorithms implemented in the OpenCV library. The calibration procedure does not only rely on these functions but is also extended by own approaches, for instance the computation of the disparity-to-depth mapping matrix. With the presented calibration procedure it is possible to calibrate an array of

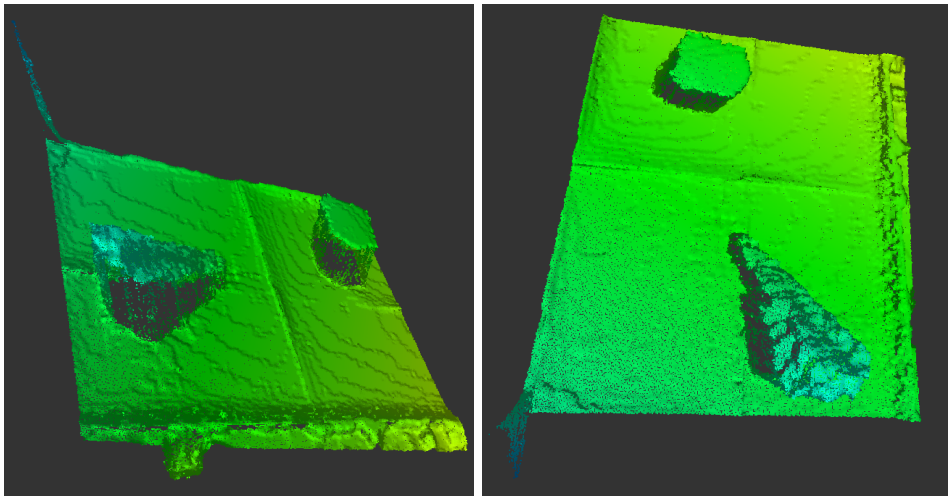

Figure 8. Two views of the point cloud resulting from the reprojection of the disparity map to 3D world coordinates. Only the valid area (cf. Figure 7) is depicted.

cameras, segregated into stereoscopic camera pairs, and retrieve 3D information about the contemplated scene.

In the future the individual point clouds will be registered and filtered to obtain one point cloud containing all information from all views. Another significant step is the usage of structured light in order to make the SGBM algorithm work on all parts of the acquired images.

ACKNOWLEDGMENTS

This paper was supported in part by the European Social Fund and Thüringer Aufbaubank.

REFERENCES

[1] Wilburn, B., Joshi, N., Vaish, V., Levoy, M., and Horowitz, M., "High-speed videography using a dense camera array," *CVPR'04 Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2004).

[2] Vaish, V., [*Synthetic Aperture Imaging using dense Camera Arrays*], Stanford University (2007).

[3] Wang, Y., Li, Y., and Zheng, J., "A camera calibration technique based on opencv," *The 3rd International Conference on Information Sciences and Interaction Sciences* (2010).

[4] Hartley, R. and Zisserman, A., [*Multiple View Geometry in Computer Vision*], Cambridge University Pr. (2003).

[5] Hartley, R., "Theory and practice of projective rectification," *International Journal of Computer Vision* (1999).

[6] Zhang, Z., "A flexible new technique for camera calibration," *Pattern Analysis and Machine Intelligence, IEEE Transactions* (2000).

[7] Bouget, J.-Y., "Camera calibration tool box for matlab." Caltech Vision 2015, http://www.vision.caltech.edu/bouguetj/calib_doc/. (Accessed: 15 April 2018).

[8] Hirschmueller, H., "Stereo processing by semiglobal matching and mutual information," *Pattern Analysis and Machine Intelligence, IEEE Transactions* (2008).

[9] Birchfield, S. and Tomasi, C., "A pixel dissimilarity measure that is insensitive to image sampling," *Pattern Analysis and Machine Intelligence, IEEE Transactions* (1998).