Ground-truth Tracking Data Generation using Rotating Real-World Objects

Zoltán Pusztai[2,1] and Levente Hajder[1]

[1] Distributed Events Analysis Research Laboratory, MTA SZTAKI
Kende utca 13-17. H-1111 Budapest, Hungary
{pusztai.zoltan,hajder.levente}@sztaki.mta.hu,
http://web.eee.sztaki.hu/
[2] Eötvös Loránd University Budapest, Hungary

Abstract. Quantitative comparison of feature matchers/trackers is essential in 3D computer vision as the accuracy of spatial algorithms mainly depends on the quality of feature matching. This paper shows how a structured-light applying turntable-based evaluation system can be developed. The key problem here is the highly accurate calibration of scanner components. The ground truth (GT) tracking data generation is carried out for seven testing objects. It is shown how the OpenCV3 feature matchers can be compared on our GT data, and the obtained quantitative results are also discussed in detail.

1 INTRODUCTION

Developing a realistic 3D approach for feature tracker evaluation is very challenging since realisticly moving 3D objects can simultaneously rotate and translate, moreover, occlusion can also appear in the images. It is not easy to implement a system that can generate ground truth (GT) data for real-world 3D objects. The aim of this paper is to present a novel structured-light reconstruction system that can produce highly accurate feature points of rotating spatial objects.

The Middlebury database[3] is considered as the state-of-the-art GT feature point generator. The database itself consists of several datasets that had been continuously developed since 2002. In the first period, they generated corresponding feature points of real-world objects (Scharstein and Szeliski, 2002). The first Middlebury dataset can be used for the comparison of feature matchers. Later on, this stereo database was extended with novel datasets using structured-light (Scharstein and Szeliski, 2003) or conditional random fields (Pal et al., 2012). Even subpixel accuracy can be achieved in this way as it is discussed in (Scharstein et al., 2014).

However, our goal is to generate tracking data via multiple frames, the stereo setup is too strict limitation for us. The description of the optical flow datasets of Middlebury database was published in (Baker et al., 2011). It was developed in order to make the optical flow methods comparable. The latest version contains four kinds of video sequences:

[3] http://vision.middlebury.edu/

1. *Fluorescent images*: Nonrigid motion is taken by a color and a UV-camera. Dense ground truth flow is obtained using hidden fluorescent texture painted on the scene. The scenes are moved slowly, at each point capturing separate test images in visible light, and ground truth images with trackable texture in UV light.
2. *Synthesized database*: Realistic images are generated by an image syntheses method. The tracked data can be computed by this system as all parameters of the cameras and the 3D salcene are known.
3. *Imagery for Frame Interpolation.* GT data is computed by interpolating the frames. Therefore the data is computed by a prediction from the measured frames.
4. *Stereo Images of Rigid Scenes.* Structured light scanning is applied first to obtain stereo reconstruction. (Scharstein and Szeliski 2003). The optical flow is computed from ground truth stereo data.

The main limitation of the Middlebury optical flow database is that the objects move approximately linearly, there is no rotating object in the datasets. This is a very strict limitation as tracking is a challenging task mainly when the same texture is seen from different viewpoint.

It is interesting that the Middlebury multi-view database (Seitz et al., 2006) contains ground truth 3D reconstruction of two objects, however, the ground truth tracking data were not generated for these sequences. Another limitation of the dataset is that only two low-textured objects are used.

It is obvious that tracking data can also be generated by a depth camera (Sturm et al., 2012) such as Microsoft Kinect, but its accuracy is very limited. There are other interesting GT generators for planar objects such as the work proposed in (Gauglitz et al., 2011), however, we would like to obtain the tracked feature points of real spatial objects.

Due to these limitations, we decided to build a special hardware in order to generate ground truth data. Our approach is based on a turntable, a camera, and a projector. They are not too costly, however, the whole setup is extremely accurate as it is shown here.

Accurate calibration of turntable-based 3D scanners. The application of structured-light scanner is a relatively cheap and accurate solution to build a real 3D scanner as it is discussed in the latest work of (Moreno and Taubin, 2012). Another possibility for a very accurate 3D reconstruction is laser scanning (Bradley et al., 1991), however, the accurate calibration of the turntable is not possible using a laser stripe since it can only reconstruct a 2D curve at a moment. For turntable calibration, the reconstruction of 3D objects is a requirement since the axis of the rotation can be computed by registrating the point clouds of the same rotating object.

Moreover, the calibration of the camera and projector intrinsic and extrinsic parameters is also crucial. While the camera calibration can be accurately carried out by the well-known calibration method of (Zhang, 2000), the projector calibration is a more challenging task. The projector itself can be considered as an inverse camera: while the camera projects the 3D world to the 2D image, the projector projects the planar image onto the 3D world. For this reason, the corresponding points of the 3D world and the projector image cannot be matched. Therefore, firstly the pixel-pixel correspondences have to be detected between the camera and the projector. The application of structured light was developed in order to efficiently realize this correspondence detection (Scharstein and Szeliski, 2003).

Many projector calibration methods exist in the field. The first popular class of existing solutions (Sadlo et al., 2005; Liao and Cai, 2008; Yamauchi et al., 2008) is to (i) use a calibrated camera to determine the world coordinate, (ii) then a pattern is projected onto the calibration plane, the corners are detected and locations are estimated in 3D, (iii) the 3D → 2D correspondences are given by running the (Zhang, 2000) calibration. The drawback of this kind of approaches is that its accuracy is relatively low since the projected 3D corner locations are estimated, and these estimated data are used for the final calculation.

Another possible solution is to ask the user to move the projector at different positions (Anwar et al., 2012; Jamil Draréni, 2009). It is not possible for our approach as the projector is fixed. Moreover, the accuracy of these kind of approaches is also low.

There are algorithms where both projected and printed pattern are used (Audet, 2009; Martynov et al., 2011). The main idea here is that if the projected pattern is iteratively adjusted until it fully overlaps the printed pattern, then the projector parameters can be estimated. Color patterns can also be applied for this purpose (Park and Park, 2010). However, we found that this quite complicated method is not required to calibrate the camera-projector system.

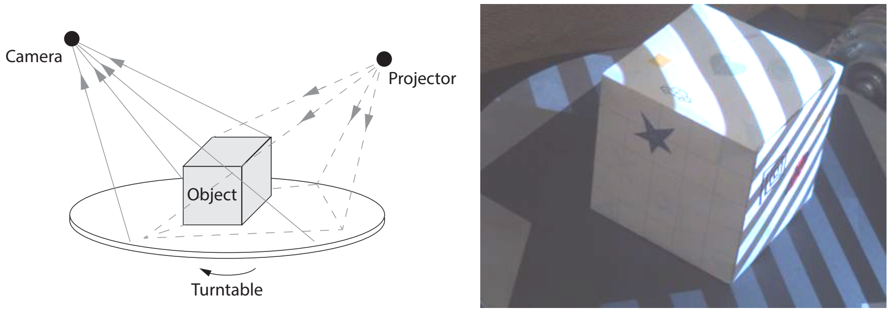

Fig. 1: Left: Setup of structured light scanning. Right: Camera snapshot of our realized scanner.

Our calibration methods for both the camera and projector use a simply chessboard plane. Our algorithms are very similar to those of (Moreno and Taubin, 2012). As it is shown later, we calibrate the camera first by the method of (Zhang, 2000). Then the point correspondences between camera and projector pixels are determined by robustly estimating the local homography close to the chessboard corners. The intrinsic projector parameters can be computed by (Zhang, 2000) as well. The extrinsic parameters (relative translation and orientation between the camera and the projector) can given by a stereo calibration problem. For this purpose, there are several solutions as it is discussed in (Hartley and Zisserman, 2003) in detail. However, we found that the accuracy of stereo calibration is limited, therefore we propose a more sophisticated estimation here.

Contribution of this study. The main novelty of this paper is that we show here how very accurate GT feature data can be generated for rotating object if a camera-projector system is applied with turntable. To the best of our knowledge, our approach is the first system that can yield such accurate GT tracking data. The usage of a turntable for 3D reconstruction itself is not a novel idea, but its application for GT data generation it is. The calibration algorithms within the system have a minor and a major improvements:

- The camera-projector correspondence estimation is based on a robust (RANSAC-based) homography estimation.
- The turntable calibration is totally new: while usual turntable calibrators (Kazo and Hajder, 2012) compute the axis by performing a usual chessboard-based calibration method (Zhang, 2000) for the rotating chessboard plane, and the axis of the rotation is computed from the extrinsic camera parameters, we propose a novel optimization problem that minimizes the reprojection for the corners of the rotating chessboard. We found the accuracy of this novel algorithm is significantly higher. During the turnable calibration, the extrinsic parameters of the camera and projector are also obtained.

Another significant contribution of our study is that the feature matchers implemented in OpenCV3 are quantitatively compared on our GT data.

Structure of the paper. First we deal with the calibration of the components: the camera, projector and turntable calibration is described one by one in Secs. 2.1, 2.2, and 2.3, respectively. Sec. 3 shows how accurate GT data can be generated by the developed equipment. Feature matchers of OpenCV3 are compared on this novel GT dataset. Finally, Sec. 4 concludes the work and discusses the limitations.

2 PROPOSED EQUIPMENT AND ALGORITHMS

Our 3D scanner consists of 3 main parts. It is visualized in Fig. 1. The left plot is the schematic setup, while the right one shows a snapshot of the camera when the object is illuminated by the structured light. The main components of the equipment are the camera, the projector, and the turntable. Each of the above needs to be calibrated correctly to reach high accuracy in 3D scanning. The camera and the projector are fixed to their arms, but the turntable can move[4]: it is able to rotate the object to be reconstructed.

The bottleneck of the proposed approach is the calibration of the components. In this section, it is overviewed how the camera, the projector, and the axis of the rotating table can be accurately calibrated.

2.1 Camera Calibration

For representing the camera, we choose the pinhole model with radial distortion. Assuming that the coordinate system is aligned to the camera, the projection of the point

[4] These arms are also moving, but their calibration is not considered here, it is a possible future work.

$X \in \mathbb{R}^3$ onto the camera plane is $\tilde{u} = [\tilde{u}_x, \tilde{u}_y, 1]^T \in \mathbb{R}^3$, which can be described by the following equation:

$$\tilde{u} \sim K_c X, \quad K_C = \begin{bmatrix} f_x & \gamma & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix},$$

where K_C stands for the camera matrix, f_x and f_y are the focal length, (p_x, p_y) is the principal point, and γ is the shear, while the operator $\sim$ denotes equality up to a scale.

Furthermore, the standard radial distortion model (Hartley and Zisserman, 2003) is also used to obtain the distorted 2D coordinates in vector $u \in \mathbb{R}^2$. In our case, we use only radial distortion which can be described by two parameters: (k_1, k_2). The applied distortion model here is as follows:

$$u = \begin{bmatrix} \tilde{u}_x[1 + k_1 r^2 + k_2 r^4] \\ \tilde{u}_y[1 + k_1 r^2 + k_2 r^4] \end{bmatrix}, \quad r^2 = \tilde{u}_x^2 + \tilde{u}_y^2$$

The camera matrix and the distortion parameters are together called the intrinsic parameters of the camera.

A black and white chessboard is held in sight of the camera at arbitrary positions. Images were taken and the chessboard corners were found on the images, and they refined to reach sub-pixel precision. Then we can compute the intrinsic parameters of the camera by the method of (Zhang, 2000).

2.2 Projector Calibration

Since the projector can be viewed as an inverse camera, it can be represented by the same model applied for the camera before. However, finding the right projector pixels, which through the chessboard corners are seen from the viewpoint of the projector is not so obvious. To overcome this problem, a structured light sequence is projected to the scene. It precisely encodes the pixel coordinates in the projector image. For each scene point, the projected codes has to be decoded. From now, the chessboard has to be placed in a position that can be viewed from both the camera and projector.

The structured light we used for the calibration is based on the binary Gray code since it is the most accurate coding for structured light scanning as it is discussed in (Scharstein and Szeliski, 2003). In addition, we project inverse images after every single one, meaning that every pixel on the images is reversed. But before structured light utilized, full black and white images are projected for easier object recognition, and for easier decoding of the structured light.

Since the resolution of our projector is 1024 × 768, the number of the projected images are 42 for each chessboard orientation. The projected sequence consists of 2 pure black and white images, 10 images for encoding the horizontal, and 10 for encoding the vertical coordinates of each projector pixel. Additionally, the inverse images have to be inserted into the sequence as well. These images are taken from one viewpoint, and they are called as the image set.

After all the images are taken, one can begin the decoding of the structured light. First of all, we calculate the direct and indirect intensity of the light, pixel by pixel for each image set. The full method is described in (Nayar et al., 2006). Then the minimum and maximum intensities are determined per pixel and the direct and indirect values are given by the equations as follows:

$$L_D = \frac{L_{max} - L_{min}}{1-B}, \quad L_I = \frac{2(L_{min} - B * L_{max})}{1-B^2},$$

where B is the amount of light emitted by a turned-off projector pixel. We needed to separate these two components from each other, because we are only interested in the direct intensities illuminated by the projector.

Then we need to classify the pixels on each image pair, consisting the image given by the structured light and its inverse. There are 3 clusters to classify into: (1) The pixel is illuminated on the first image. (2) The pixel is not illuminated on the first image. (3) Cannot be determined.

The classification rules are as follows:

- $L_D < M \implies$ the pixel is in the third class,
- $L_d > L_I \wedge P_1 > P_2 \implies$ the pixel is illuminated,
- $L_d > L_I \wedge P_1 < P_2 \implies$ the pixel is not illuminated,
- $P_1 < L_D \wedge P_2 > L_I \implies$ the pixel is not illuminated,
- $P_1 > L_I \wedge P_2 < L_I \implies$ the pixel is illuminated,
- otherwise it cannot be determined.

The pixel intensity in the first and inverse images are denoted by P_1, and P_2, while M is a user-defined threshold: $M = 5$ is set in our approach.

For further reading about the classification, we recommend to read the study of (Xu and Aliaga, 2007).

Since the chessboard consists of alternating black and white squares, decoding near the chessboard corners can resolve errors. To avoid these errors, we calculate local homographies around the corners. We use 11 pixel-wide kernel window and every successfully decoded projector pixel is in consideration. For the homography estimation, a RANSAC-based (Fischler and Bolles, 1981) DLT homography estimator is applied in contrast to the work of (Moreno and Taubin, 2012) where robustification is not dealt with. We found that the accuracy is increased when RANSAC-scheme is applied. After the homography is computed among the camera and projector pixels, we use this homography to transform the camera pixels to the projector image. In this way we get the exact projector pixels we needed, so we can use the method of (Zhang, 2000) to calibrate the projector. Remark that the extrinsic projector calibration is refined later, but the intrinsic parameters are not.

2.3 Turntable Calibration

The aim of the turntable calibration is to compute the axis of the turntable. It is represented by a point and a direction. Therefore, the degree of freedom of a general axis estimation is four (2 DoFs: position of a plane; other 2 DoFs: direction) .

Fortunately, the current problem is constrained. We know that the axis is perpendicular to the plane of the turntable. Thus, the direction is given, only the position should be calculated within the turntable plane.

The turntable is calibrated if we know the centerline around which the table is turning. Two methods was used to calculate this 3D line. First we place the chessboard on the turntable, and start rotating it. Images are taken between the rotations, and the extrinsic parameters can be computed for each image since the camera is already calibrated. This motion is equivalent with the motion of a fixed chessboard and a moving camera. The circle that the camera follows has the same centerline as the turntable. Thus fitting a circle to the camera points gives the centerline of the turntable (Kazo and Hajder, 2012).

However, we found that this method is not accurate enough. Therefore, we developed a novel algorithm that is overviewed in the rest of this section.

Problem statement of turntable axis calibration Given a chessboard with known size, for which the corners can be easily detected by widely used pattern recognition algorithms, the goal is to estimate the axis of the turntable. This is part of the calibration of a complex structured-light 3D reconstruction system that consists of one camera, one projector, and one turntable. The latter one is driven by a stepping motor, the angle of the rotation can be very accurately set. The camera and projector intrinsic parameters are also known, in other words, they are calibrated.

The input data for axis calibration comes from detected chessboard corners. The chessboard is placed on the turntable. Then it is rotated and images are taken with different rotational axis. The corners are detected on all of these images. The chessboard is placed in a higher position on the turntable, but the new plane orientation is also parallel to the turntable. Then the chessboard is rotated, and the corners are detected as well. (The chessboard can be placed in arbitrary altitudes. We only use two different values, but the proposed calibration method can work with arbitrary number of positions.)

If we consider the case when the planes of the chessboard and the turntable are parallel, the distance between them is h, then the chessboard corners can be written as

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} x - o_x \\ y - o_y \end{bmatrix} + \begin{bmatrix} o_x \\ o_y \end{bmatrix} = \begin{bmatrix} x\cos\alpha - y\sin\alpha + o_x(1-\cos\alpha) + o_y\sin\alpha \\ x\sin\alpha + y\cos\alpha - o_x\sin\alpha + o_y(1-\cos\alpha) \end{bmatrix}, \tag{1}$$

where α denotes the current angle of the rotation. Note that altitude h does not influence the relationship. Also remark that capital X and Y denote spatial coordinates, while their lowercase letters (x and y) are 2D coordinates in image space.

Proposed algorithm The proposed axis calibration consists of two main steps:

1. Determination of the axis center $[o_x, o_y]^T$ on chessboard plane, and
2. computation of the camera and projector extrinsic parameters.

Axis center $[o_x, o_y]^T$ estimation on chessboard plane. The goal of the axis center estimation is to calculate the location $[o_x, o_y]^T$. We propose an alternation-type method with two substeps:

Homography-step. The plane-plane homography is estimated for each image. The 2D locations of the corners in the images are known. The 2D coordinates can be determined in the chessboard plane by Eq. 1. If the homogenous coordinates are used, the relationship becomes

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \sim H \begin{bmatrix} x\cos\alpha - y\sin\alpha + o_x(1-\cos\alpha) + o_y\sin\alpha \\ x\sin\alpha + y\cos\alpha - o_x\sin\alpha + o_y(1-\cos\alpha) \\ 1 \end{bmatrix}. \tag{2}$$

We apply the standard normalized direct linear transformation (normalized DLT) with a numerical refinement step (Hartley and Zisserman, 2003) in order to estimate the homography. It minimizes the linearized version of Eq. 2:

$$E(\alpha, x, y, o_x, o_y) = E_1(\alpha, x, y, o_x, o_y) + E_2(\alpha, x, y, o_x, o_y),$$

where

$$\begin{aligned} E_1(\alpha, x, y, o_x, o_y) = {} & uh_{31}(x\cos\alpha - y\sin\alpha + o_x(1-\cos\alpha) + o_y\sin\alpha) + \\ & uh_{32}(x\sin\alpha + y\cos\alpha - o_x\sin\alpha + o_y(1-\cos\alpha)) + uh_{33} - \\ & h_{11}(x\cos\alpha - y\sin\alpha + o_x(1-\cos\alpha) + o_y\sin\alpha) - \\ & h_{12}(x\sin\alpha + y\cos\alpha - o_x\sin\alpha + o_y(1-\cos\alpha)) - h_{13}, \end{aligned}$$

and

$$\begin{aligned} E_2(\alpha, x, y, o_x, o_y) = {} & vh_{31}(x\cos\alpha - y\sin\alpha + o_x(1-\cos\alpha) + o_y\sin\alpha) + \\ & vh_{32}(x\sin\alpha + y\cos\alpha - o_x\sin\alpha + o_y(1-\cos\alpha)) + vh_{33} - \\ & h_{21}(x\cos\alpha - y\sin\alpha + o_x(1-\cos\alpha) + o_y\sin\alpha) - \\ & h_{22}(x\sin\alpha + y\cos\alpha - o_x\sin\alpha + o_y(1-\cos\alpha)) - h_{23}. \end{aligned}$$

This is a linear problem. The center and the scale of the applied coordinate system can be arbitrary chosen. As it is discussed in (Hartley and Zisserman, 2003), the mass center and quasi-uniform scale is the most accurate choice. The error function $E(\alpha, x, y, o_x, o_y)$ can be written for every chessboard corner point for every rotational angle. Therefore, the minimization problem is formulated as

$$\arg\min_H \sum_{i=1}^{G_x} \sum_{j=1}^{G_y} \sum_{k=1}^{N} E(\alpha_k, x_{i,\alpha}, y_{j,\alpha}, o_{x,\alpha}, o_{y,\alpha}).$$

where $a_k \in [0, 2\pi]$, $x_i \in [0, G_x]$, and $y_i \in [0, G_y]$, and G_x,G_y are the dimensions of chessboard corners, respectively. (Possible values for (x_i, y_j) are $(1,1)$,$(1,2)$,$(2,1)$,...etc.) This problem remains an over-constrained homogeneous linear one that can be optimally solved.

Axis-step. Its goal is to estimate the axis location $[o_x, o_y]^T$. The above two equations are linear with respect to the center coordinates. Therefore, the equations form a homogeneous linear system of equations $A\left[o_x, o_y\right]^T = b$, where $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ and

$$\begin{aligned}
a_{11} &= h_{11} - h_{11}\cos\alpha - h_{12}\sin\alpha - u(h_{31} - h_{31}\cos\alpha - h_{32}\sin\alpha), \\
a_{12} &= h_{11}\sin\alpha + h_{12} - h_{12}\cos\alpha - u(h_{31}\sin\alpha + h_{32} - h_{32}\cos\alpha), \\
a_{21} &= h_{21} - h_{21}\cos\alpha - h_{22}\sin\alpha - v(h_{31} - h_{31}\cos\alpha - h_{32}\sin\alpha), \\
a_{22} &= h_{21}\sin\alpha + h_{22} - h_{22}\cos\alpha - v(h_{31}\sin\alpha + h_{32} - h_{32}\cos\alpha).
\end{aligned}$$

Furthermore, $b = \left[b_{11} - b_{12}\ ,\ b_{21} - b_{22} \right]^T$, where

$$\begin{aligned}
b_{11} &= h_{13} + h_{11}(x\cos\alpha - y\sin\alpha) + h_{12}(y\cos\alpha + x\sin\alpha), \\
b_{12} &= u\left(h_{33} + h_{31}(x\cos\alpha - y\sin\alpha) + h_{32}(y\cos\alpha + x\sin\alpha\right), \\
b_{21} &= h_{23} + h_{21}(x\cos\alpha - y\sin\alpha) + h_{22}(y\cos\alpha + x\sin\alpha), \\
b_{22} &= v\left(h_{33} + h_{31}(x\cos\alpha - y\sin\alpha) + h_{32}(y\cos\alpha + x\sin\alpha)\right).
\end{aligned}$$

The above equations can be written for all corners of the chessboard for all rotated positions. Therefore, both the homography- and the axis-steps are extremely overconstrained, thus the parameters can be very accurately estimated. It is interesting that the homography and the axis location estimations are homogeneous, and inhomogeneous linear problems, respectively. They can also solved for the over-determined case using the Moore-Penrose pseudo-inverse of the coefficient matrix as it is well-known (Björck, 1996).

The two substeps have to be run one after the other. Both steps minimize the same algebraic error, therefore the method converges to the closest (local) minimum. Unfortunately, global optimum cannot be theoretically guaranteed. But we found that the algorithm converges to the correct solution. The speed of the convergence is relatively fast, to our experiments, $20-30$ iterations are required to reach the minimum.

Parameter Initialization. The proposed alternation method requires initial values for o_x and o_y. It has been found that the algorithm is not too sensitive to the locations of the initial values. The center of the chessboard is an appropriate solution for o_x and o_y. Moreover, we have tried more sophisticated methods. If the camera centers are estimated by a Perspective n Point (PnP) algorithm such as (Lepetit et al., 2009), then the camera centers for the rotating sequence form a circle (Kazo and Hajder, 2012) as it is mentioned in the first part of this section. The center of this circle is also a good initial value. However, we found that the correct solution is reached as well if the initial center is an arbitrary point within the chessboard region.

Axis center estimation in the global system The first algorithm estimates the center of the axis in the coordinate system of the chessboard. But the chessboard are placed in different positions with different altitudes. The purpose of the algorithm discussed in this section is to place the rotated chessboard in the global coordinate system and to determine the extrinsic parameters (location and orientation) of the projector. The global system is fixed to the camera, therefore, the camera extrinsic parameters have not to be estimated.

In our calibration setup, only two chessboard sequences are taken. The extrinsic position can be easily determined. If the 3D coordinates of the plane are known, and the 2D locations are detected, then the estimation of the projective parameters is called the PnP problem. Mathematically, the PnP optimization can be written as

$$\arg\min_{R,t} \sum_{i=1}^{G_x} \sum_{j=1}^{G_y} \sum_{k=1}^{N} Rep\left(R, t, \begin{bmatrix} u_{i,\alpha} \\ v_{j,\alpha} \end{bmatrix}, \begin{bmatrix} x'_{i,\alpha} \\ h \end{bmatrix}\right)$$

where the definition of the function *Rep* is as follows:

$$Rep\left(R, t, \begin{bmatrix} u_i \\ v_j \end{bmatrix}, \begin{bmatrix} x'_i \\ y'_j \\ h \end{bmatrix}\right) = \left\| DeHom\left(R \begin{bmatrix} x'_i \\ y'_j \\ h \end{bmatrix} + t\right) - \begin{bmatrix} u_i \\ v_j \end{bmatrix} \right\|_2^2 .$$

The applied comma (’) means that the origin of the coordinate system for chessboard corners are placed at $[o_x, o_y]^T$. Function *DeHom* gives the dehomogeneous 2D vector of a spatial vector as $DeHom([X,Y,Z]^T) = [X/Z, Y/Z]^T$.

There are PnP methods that can cope with planar points. We used the EPnP (Lepetit et al., 2009) algorithm for our approach. At this point, the relative transformation between the chessboard planes and the camera can be calculated. They are denoted by $[R^1, t^1]$, and $[R^2, t^2]$. The altitude of the chessboard can be measured. Without loss of generalization, altitude of the first plane can be set to zero: $h_1 = 0$. (The simplest way is to set the first chessboard to the turntable. Then the altitude of the second chessboard can be easily measured with respect to the turntable.)

The final task is to calculate the relative angle $\Delta\alpha$ between the two rotating chessboard planes.The estimation of one parameter is relatively simple. We solve it by exhaustive search. The best value is given by the rotation for which the reprojection error of the PnP problem is minimal:

$$\arg\min_{R,t} \sum_{i=1}^{G_x} \sum_{j=1}^{G_y} \sum_{k=1}^{N} \left(Rep^1 + Rep^2\right)$$

where

$$Rep^1 = Rep\left(R^1, t^1, \begin{bmatrix} u_{i,\alpha_k} \\ v_{j,\alpha_k} \end{bmatrix}, \begin{bmatrix} x'_{i,\alpha_k} \\ y'_{j,\alpha_k} \\ 0 \end{bmatrix}\right), Rep^2 = Rep\left(R^2, t^2, \begin{bmatrix} u_{i,k} \\ v_{i,k} \end{bmatrix}, \begin{bmatrix} x'_{i,\alpha+\alpha_k} \\ y'_{j,\alpha+\alpha_k} \\ h \end{bmatrix}\right).$$

The upper index denotes the number of the chessboard. The relationship between the left and right terms are that the spatial points have to rotated with the same angle, but a fix angular offset $\Delta\alpha$ has to be added to each rotation for the second chessboard plane with respect to the first one. The impact of $\Delta\alpha$ for the second rotation matrix is written as follows:

$$R^2 = \begin{bmatrix} \cos\Delta\alpha & -\sin\Delta\alpha & 0 \\ \sin\Delta\alpha & \cos\Delta\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} R^1 \tag{3}$$

The minimization problem is also a PnP one, therefore it can be efficiently solved by the algorithm of (Lepetit et al., 2009). The estimation of $\Delta\alpha$ is obtained by an exhaustive search.

Finally, the extrinsic parameters of the projector are computed by running the PnP algorithm again for the corners detected in the projector images. The obtained projector parameters have to be transformed by the inverse of the camera extrinsic parameters since our global coordinate system is fixed to the camera.

2.4 Object Reconstruction

The object reconstruction looks very similar to the projector calibration. In this case, an object is placed on the turntable instead of the chessboard. Structured light is projected onto it, images are taken, then the object is rotated. This procedure is repeated until the object returns to the starting position. Then we decode the projector pixels from the projected structured light in each image set. After it is done, we use the Hartley-Strum triangulation technique (Hartley and Sturm, 1997) for corresponding camera-projector pixels due to its accuracy to determine the object points from one viewpoint. We calculate these for each viewpoint, and then we can combine the point sets together, which results a 3D points set of the full object.

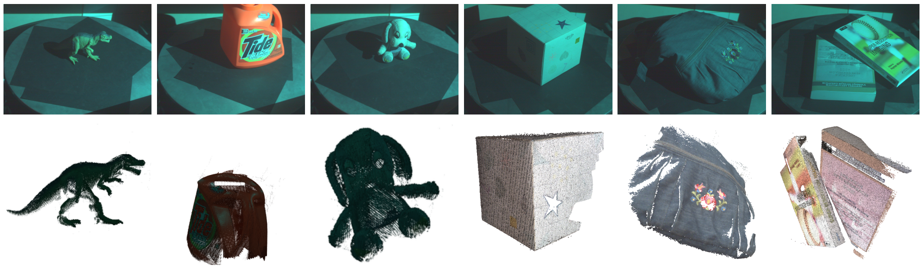

Fig. 2: Top row: Camera snapshots for the input sequences. Bottom: Reconstructed 3D point clouds. Test sequences from left to right: ’Dinosaur’, ’Flacon’, ’Plush dog’, ’Cube’, ’Bag’, and ’Books’.

3 Comparison of Feature Matchers Implemented in OpenCV3

The main advantage of our method is that the whole GT data generation is totally automatic. Therefore, arbitrary number of objects can be reconstructed. We show here seven typical objects that have well trackable feature points. They are as follows:

- **Dinosaur**. A typical computer vision study deals with the reconstruction of a dinosaurs as it is shown in several scientific papers, e.g (”Fitzgibbon et al., 1998). It has a simple diffuse surface that is easy to reconstruct in 3D, hence the feature matching is possible. For this reason, a dino is inserted to our testing dataset.

- **Flacon.** The plastic holder is another smooth and diffuse surface. A well-textured label is fixed on the surface.
- **Plush Dog.** The tracking of the feature point of a soft toy is a challenging task as it does not have a flat surface. A plush dog is included into the testing database that is a real challenge for feature trackers.
- **Poster.** The next sequence of our dataset is a rotating poster coming from a motorcycle magazine. It is a relatively easy object for feature matchers since it is a well-textured plane. The pure efficiency of the trackers can be checked in this example due to two reasons: (i) there is no occlusion, and (ii) the GT feature tracking is equivalent to the determination of plane-plane homographies.
- **Cube.** This object is manually textured and its material is cardboard. The texturing is quite sparse, only a little part of the object area can be easily tracked.
- **Bag.** A canvas bag with beautiful and well-textured decoration is a good test example. It seems to be as easy as the Poster, however, its shape is more varied.
- **Books.** Different books are placed on the turntable. The book covers are well-textured.

During the test, the objects were rotated by the turntable, the difference of the degree of two subsequent was set to 3°. Our GT tracking data generator has two modes. (i) The first version regularly generates the feature points in the first image. The feature points are located across a regular grid in the valid region of the camera image. (ii) The points in the first image is determined by the applied feature generator.

Then the generated feature points were reconstructed in the first image using the structured light. These 3D reconstructed point coordinates were rotated around the turntable axis with the known rotating axis, and projected to the next image. This procedure was repeated for all the images of the test sequence. The 2D feature coordinates after projection give the final GT for quantitative feature tracker comparison.

Input images and the corresponding (reconstructed) 3D point clouds of all testing sequences except ’Poster’ are visualized in Figure. 2. Sequence ’Poster’ is missing as it is a planar paper and its 3D model is not interesting. The 3D models are represented by colored point clouds, however, the color itself does not influence the reconstruction. It is only painted due to its spectacularity.

The computed ground truth data for the sequence ’Poster’ are pictured in Fig. 3. The first row shows the tracked points when the points are selected across a grid. The second row of Fig. 3 consist of images with the tracked GT SIFT feature points (yellow dots).

The obtained ground truth data were visually checked by us and we have not found any inaccuracy on it. We think that the accuracy is below pixel, in other word, subpixel accuracy was reached. This is extremely low as the camera resolution is 2592×1936 (5 Mpixel).

Compared methods. Firstly, the possibilities is overviewed that OpenCV can give about feature tracking. The currently supported feature detectors in OpenCV are as follows: AGAST (Mair et al., 2010), AKAZE (Pablo Alcantarilla (Georgia Institute of Technolog), 2013), BRISK (Leutenegger et al., 2011), FAST (Rosten and Drummond, 2005), GFTT (Tomasi, C. and Shi, J., 1994) (Good Features To Track – also known as Shi-Tomasi corners), KAZE (Alcantarilla et al., 2012), MSER (Matas et al., 2002), ORB (Rublee et al., 2011).

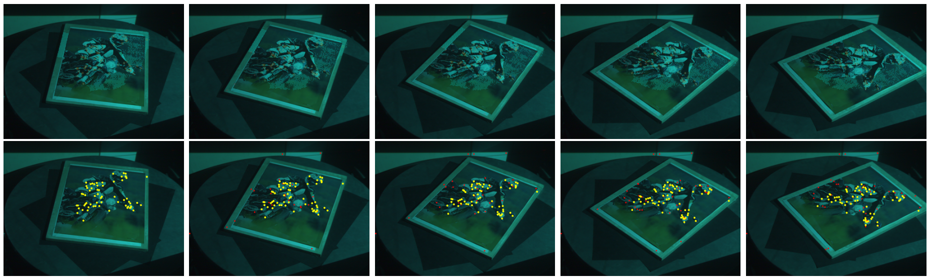

Fig. 3: The visualized ground truth tracking data drawn on images of the 'Poster' sequence. Top row: features generated by a grid within valid image region. Bottom row: features generated by SIFT method. The images are best viewed in color.

However, if one compiles the contrib(nonfree) repository with the OpenCV, the following detectors can also be used: SIFT (Lowe, 1999b), STAR (Agrawal and Konolige, 2008), and SURF (Bay et al., 2008).

We use our scanner to take 20 images of a rotating object. After each image taken, a structured light sequence is projected in order to make the reconstruction available for every position. (Reconstructing the points only in the first image is not enough.)

Then we start searching for features in these images using all feature detectors. After the detection is completed, it is required to extract descriptors. Descriptors are needed for matching the feature points in different frames. The following descriptors are used (each can be found in OpenCV): AKAZE (Pablo Alcantarilla (Georgia Institute of Technolog), 2013), BRISK (Leutenegger et al., 2011), KAZE (Alcantarilla et al., 2012), ORB (Rublee et al., 2011). If one compiles the contrib repository, he/she can also get SIFT (Lowe, 1999b), SURF (Bay et al., 2008), BRIEF (Calonder et al., 2010), FREAK (Ortiz, 2012), LATCH (Levi and Hassner, 2015), DAISY (Tola et al., 2010) descriptors [5].

Another important issue is the parameterization of the feature trackers. It is obvious that the most accurate strategy is to find the best system parameters for the methods, nevertheless the optimal parameters can differ for each testing video. On the other hand, we think that the authors of the tested methods can set the parameters more accurately than us as they are interested in good performance. For this reason, the default parameter setting is used for each method, and we plan to make the dataset available for everyone and then the authors themselves can parameterize their methods.

After the detection and the extraction are done, the matching is started. Every image pair is taken into consideration, and match each feature point in the first image with one in the second image. This means that every feature point in the first image will have a pair in the second one. However, there can be some feature locations in the second image, which has more corresponding feature points in the first one, but it is also possible that there is no matching point.

[5] The BRIEF descriptor is not invariant to rotation, however, we hold it in the set of testing algorithms as it surprisingly served good results.

The matching itself is done by calculating the minimum distances between the descriptor vectors. This distance is defined by the feature tracking method used. The following matchers are available in OpenCV:

- L_2 – *BruteForce*: a brute force minimization algorithm that computes each possible matches. The error is the L_2 norm of the difference between feature descriptors.
- L_1 – *BruteForce*: It is the same as L_2 – *BruteForce*, but L_1 norm is used instead of L_2 one.
- *Hamming – BruteForce*: For binary feature descriptor (BRISK, BRIEF, FREAK, LETCH,ORB,AKAZE), the Hamming distance is used.
- *Hamming2 – BruteForce*: It is a variant of the Hamming distance. The difference between Hamming and Hamming2 is that the former considers every bit as element of the vector, while Hamming2 use integer number, each bit pair forms a number from interval $0 \dots 3$ [6].
- *Flann-Based*: FLANN (Fast Library for Approximate Nearest Neighbors) is a set of algorithms optimized for fast nearest neighbor search in large datasets and for high dimensional features (Muja and Lowe, 2009).

It is needed to point out that one can pair each feature detector with each feature descriptor but each feature matchers is not applicable for every descriptor. An exception is thrown by OpenCV if the selected algorithms cannot work together. But we try to evaluate every possible selection.

The comparison of the feature tracker predictions with the ground truth data is as follows: The feature points are reconstructed first in 3D using the images and the structured light. Then, because it is known that the turntable was rotated by 3 degrees per images, the projections of the points are calculated for all the remaining images. These projections were compared to the matched point locations of the feature trackers and the L_2 norm is used to calculate the distances.

Evaluation Methodology. The easiest and usual way for comparing the tracked feature points is to compute the summa and/or average and/or median of the 2D tracking errors in each image. This error is defined as the Euclidean distance of the tracked and GT locations. This methodology is visualized in the left plot of Fig. 4.

However, this comparison is not good enough because if a method fails to match correctly the feature points in an image pair, then the feature point moves to an incorrect location in the next image. Therefore, the tracker follows the incorrect location in the remaining frames and the new matching positions in those images will also be incorrect.

To avoid this effect, a new GT point is generated at the location of the matched point even if it is an incorrect matching. The GT location of that point can be determined in the remaining frames since that point can be reconstructed in 3D as well using the structured light scanning, and the novel positions of the new GT point can be determined using the calibration data of the test sequence.

Then the novel matching results are compared to all the previously determined GT points. The obtained error values are visualized in the right plot of Fig. 4.

[6] OpenCV's documentation is not very informative about Hamming2 distance. They suggest the usage of that for ORB features. However, it can be applied for other possible descriptors, therefore all possible combinations are tried in our tests.

The error of a feature point for the i-th frame is the weighted average of all the errors calculated for that feature. For example, there is only one error value for the second frame as the matching error can only be compared to the GT location of the feature detected in the first image. For the third frame, there are two GT locations since GT error generated on both the first (original position) and second (position from first matching) image. For the i-th image, $i-1$ error values are obtained. the error is calculated as the weighted average of those. It can be formalized as $Error_{p_i} = \sum_{n=1}^{i-1} \frac{||p_i - p'_{i,n}||_2}{i-n}$, where $Error_{p_i}$ is the error for the i-th frame, p_i the location of the tested (detected) feature, while $p'_{i,n}$ is the GT location of the feature points reconstructed from the n-th frame. The weights of the distances is $1/(i-n)$ that means that older GT points has less weights. Remark that the Euclidean (L_2) norm is chosen in order to measure the pixel distances.

If a feature point is only detected in one image and was not being followed in the next one (or was filtered out in the fundamental-matrix-based filtering step), then that point is discarded.

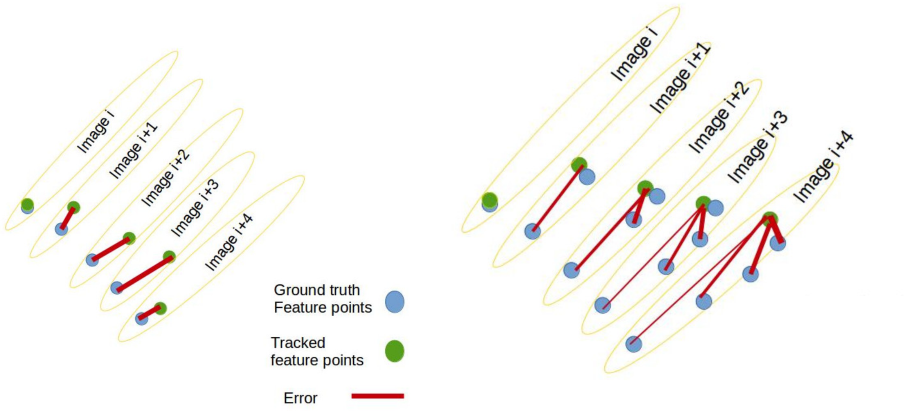

Fig. 4: Error measurement based on simple Euclidean distances(left). Applied, more sophisticated error measurement (right).

After the pixel errors are evaluated for each point in all possible images, the minimum, maximum, summa, average, and median error values of every feature points are calculated per image. The number of tracked feature points in the processed image is also counted. Furthermore, the average length of the feature tracks is calculated which shows in how many images an average feature point is tracked through.

3.1 Results

The purpose of this section is to show the main issues occurred during the testing of the feature matchers. Unfortunately, we cannot show to the Reader all the charts due to the lack of space.

General remark. The charts in this section show different combinations of detectors, descriptors, and matchers. The method 'xxx:yyy:zzz' denotes in the charts that the current method uses the detector 'xxx', descriptor 'yyy', and matcher algorithm 'zzz'.

Feature Generation and Filtering using the Fundamental Matrix The number of the detected feature points is examined first. It is an important property of the matcher algorithms since many good points are required for a typical computer vision application. For example, at least hundreds of points are required to compute 3D reconstruction of the observed scene. The matched and filtered values are calculated as the average of the numbers of generated features for all the frames as features can be independently generated in each image of the test sequences. Table 1 shows the number of the generated features and that of the filtered ones. The filtering method is based on the standard normalized RANSAC-based robust eight-point fundamental matrix extimation method implemented in OpenCV.

There are a few interesting behaviors within the data:

- The best images for feature tracking are obtained when the poster is rotated. The runner up is the sequence 'Book'. The feature generators give significantly the most points in these cases when the scenes consist of large well textured planes. It is a more challenging task to find good feature points for the rotating non-planar objects such as the dog or dinosaur. It is because the area of these objects in the images are smaller than that a book or a poster. Another interesting behavior that only a few outliers are retrieved for the sequence 'Cube' due to the lack of large well-textured areas.
- It is clearly seen that number of SURF feature points are the highest in all test cases after outlier removal. This fact suggests that they will be the more accurate features.
- The MSER method gives the most number of feature points, however, more than 90% of those are filtered. Unfortunately, the OpenCV3 library does not contain sophisticate matchers for MSER such as (Forssn and Lowe, 2007), therefore its accuracy is relatively low [7].
- Remark that the GFTT algorithm usually gives 1000 points as the maximum number was set to thousand for this method. It is a parameter of OpenCV that may be changed, but we did not modify this value.

Matching accuracy Two comparisons were carried out for the feature tracker methods. In the first test, every possible combination of the feature detectors and descriptors is examined, while the detectors are only combined with their own descriptor in the second test.

It is important to note that not only the errors of feature trackers should be compared, we must also pay attention to the number of features in the images and the feature track lengths [8]. A method with less detected features usually obtains better results (lower error rate) than other methods with higher number of features. The mostly used chart is the AVG-MED, where the average and the median of the errors are shown.

[7] Many researchers have informed us that the OpenCV MSER implementation is not perfect.

[8] Feature track length is defined as the number of images on which the feature appears.

Table 1: Mean number of generated feature points (#Fea.) and that of inliers (#Inl.). Maximal values denoted by bold font.

	Plush dog		Poster		Flacon		Dinosaur		Books		Cube		Bag	
Detector	#Fea.	#Inl.	#Fea.	#Inl.	Fea.	#Inl.	#Fea.	#Inl.	#Fea.	#Inl.	#Fea.	#Inl.	#Fea.	#Inl.
BRISK	21.7	16.9	233.6	188.8	219.7	161.0	21.6	14.8	144.2	110.2	9.75	5.8	8.3	5.2
FAST	19.7	9.4	224.8	139.2	387.0	275.4	51.1	27.0	490.3	305.2	28.8	17.9	61.0	34.6
GFTT	1000	38.2	956.7	618.8	1000	593.4	1000	92.0	1000	703.7	1000	65.5	903.3	232.0
KAZE	68.6	40.8	573.5	469.1	484.1	387.9	58.6	33.9	302.5	256.2	17	12.5	24.5	20.0
MSER	**5321**	10.6	**4864**	40.3	**3664**	31.7	**5144**	17.9	**5092**	85.6	**6062**	24.3	**4528**	41.4
ORB	42.3	34.1	259.5	230.8	337.7	287.4	67.1	45.9	253.7	206.2	28.5	23.2	52.7	41.3
SIFT	67.7	42.8	413.4	343.1	348.2	260.9	52.8	35.0	311.7	52.2	32.9	23.9	27.1	14.5
SURF	514.1	**326.0**	1877	**1578**	953.0	**726.8**	277.0	**132.6**	2048.7	**1463**	180.6	**105.4**	396.7	**261.5**
AGAST	22.5	11.8	275.8	200.3	410.2	303.5	55.0	29.9	591.3	390.5	33.1	22.7	65.8	38.2
AKAZE	144.0	101.7	815.0	761.4	655.0	553.1	89.1	59.2	282.1	260.9	18.4	15.3	23.0	20.2

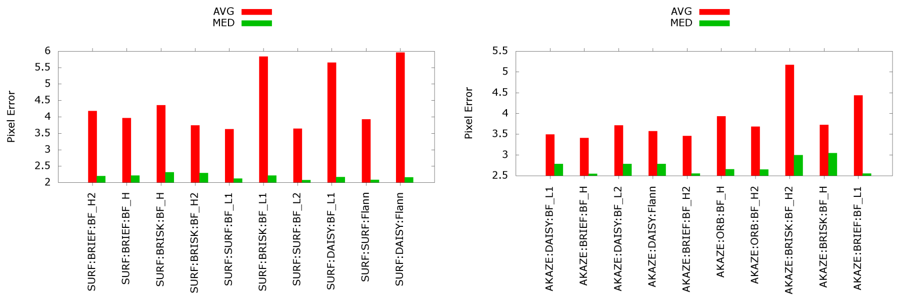

Fig. 5: Average and median errors of top 10 methods for sequences 'Plush Dog' (left) and 'Poster' (right).

Testing by all possible algorithms. As it is seen in the left plot of Fig 5 (sequence 'Plush Dog'), the SURF method dominates the chart. With the usage of SURF, DAISY, BRIEF, and BRISK descriptors more than 300 feature points remained and the median values of the errors are below 2.5 pixels, while the average is around 5 pixels. Moreover, the points are tracked through 4 images in average which yields pretty impressive statistics for the SURF detector.

The next test object was the 'Poster'. The results are visualized in the right plot of Fig 5. It is interesting to note that if the trackers are sorted by the number of the outliers and plot the top 10 methods, only the AKAZE detector remains where more than 90 percent of the feature points was considered as inlier. Besides the high number of points, average pixel error is between 3 and 5 pixels depending on the descriptor and matcher type.

In the test where the 'Flacon' object was used, we got similar results as in the case of 'Poster'. Both of the objects is rich in features, but the 'Flacon' is a spatial object. However, if we look at Fig. 6 where the methods with the lowest 10 median value were plotted, one can see that KAZE and SIFT had more feature points and can track these

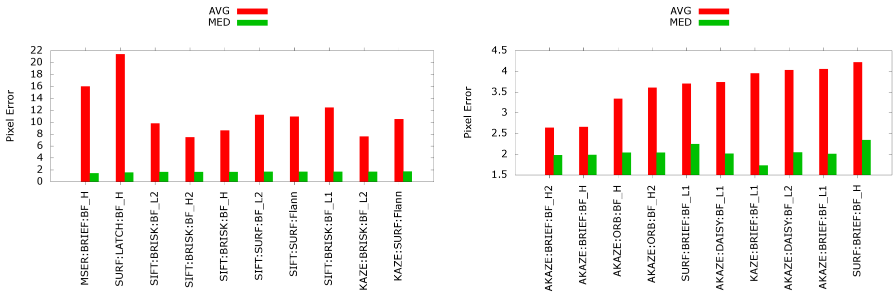

Fig. 6: Top 10 method with the lowest median for sequence ’Flacon’. Charts are sorted by median (left) and average (right) values.

over more pictures than MSER or SURF after the fundamental filtering. Even though they had the lowest median values, the average errors of these methods were rather high.

However, if one takes a look at the methods with the lowest average error, then he/she can observe that AKAZE, KAZE and SURF present in the top 10. These methods can track more points then the previous ones and the median errors are just around 2.0 pixels.

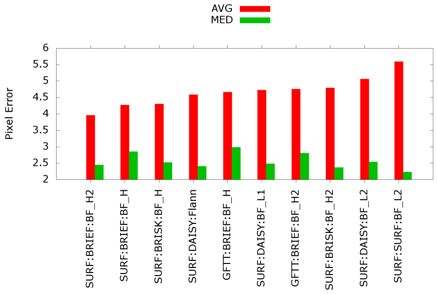

Fig. 7: Top 10 methods (with lowest average error) on sequence ’Dinosaur’.

For the sequence ’Dinosaur’ (Figure 7), the test object is very dark which makes feature detection hard. The number of available points is slightly more than 100. In this case, the overall winner of the methods is the SURF with both the lowest average and median errors. However, GFTT also present in the last chart too.

In the upper comparisons only the detectors were mentioned against each other. As one can see in the charts, most of the methods used either DAISY, BRIEF, BRISK or SURF descriptors. From the perspective of matchers, it does not really matter which type of the matcher is used for the same detector/descriptor type. However, if the descriptor gives a binary vector, then obviously the hamming distance outperforms the L2 or L1. But there are just slightly differences between the L1-L2 and H1-H2 distances. **Testing of algorithms with same detector and descriptor.** In this comparison, only the detectors that have an own descriptor are tested. Always the best matchers re selected for which the error are minimal for the observed detector/descriptor. As it can be seen in the log-scale charts in Fig. 8, the median error is almost the same for the AKAZE, KAZE, ORB and SURF trackers, but SURF is considered with the lowest average value. The tests 'Flacon' and 'Poster' result the lower pixel errors. On the other hand, the rotation of the 'Bag' was the hardest to track, it resulted much higher errors for all trackers comparing to the other tests. We think that this effect occurred because the scene contains too much non-textured parts.

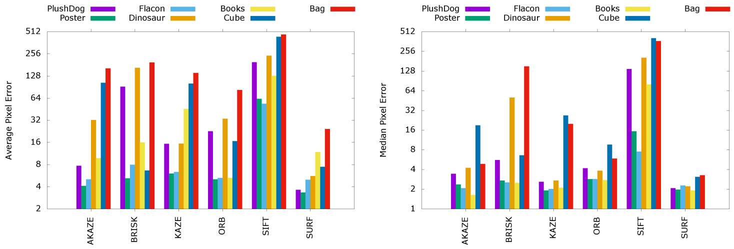

Fig. 8: Overall average (top) and median (bottom) error values for all trackers and test sequences. The detectors and descriptors were the same. Charts are best viewed in color.

4 Conclusions, Limitations, and Future Work

We have proposed a novel GT tracking data generator here that can automatically produce very accurate tracking data of rotating real-world spatial objects. The main novelty of our approach is that it consists of a turntable, and we showed how this turntable can be accurately calibrated. Finally, the validation of our equipment was shown on the quantitative comparison of OpenCV 3 matchers. It was justified that the proposed structured-light 3D scanner can produce accurate tracking data as well as realistic 3D point clouds. The GT tracking data are public, they are available at our web page [9].

The main goal of the approach proposed here is to be able to generate ground truth tracking data of real-world rotating objects. Therefore, the turntable-based equipment is unable to simulate moving cameras. However, other databases (e.g. the famous Middlebury one) can do that, thus our approach should be unified with existing datasets. Nevertheless, our equipment contains two moving arms for both the camera and projector, therefore novel viewpoints can be added to the system. It is possible if the arms are very accurately calibrated. This is a possible feature work of our GT generation project.

Another disadvantage of the current system is that part of the objects can be self-occluded due to the object rotation. This cannot be detected by the hardware, therefore

[9] http://web.eee.sztaki.hu

surface reconstruction is required to detect if the part of the scanned 3D object is occluded by another part. To avoid this problem, we plan to develop a continuous surface reconstruction method for free-form spatial objects.

Acknowledgement. This work was partially supported by the Hungarian National Research, Development and Innovation Office under the grant VKSZ_14-1-2015-0072.

[Agrawal and Konolige, 2008]Agrawal, M. and Konolige, K. (2008). Censure: Center surround extremas for realtime feature detection and matching. In *ECCV*.

[Alcantarilla et al., 2012]Alcantarilla, P. F., Bartoli, A., and Davison, A. J. (2012). Kaze features. In *ECCV (6)*, pages 214–227.

[Anwar et al., 2012]Anwar, H., Din, I., and Park, K. (2012). Projector calibration for 3d scanning using virtual target images. *International Journal of Precision Engineering and Manufacturing*, 13(1):125–131.

[Audet, 2009]Audet, S.and Okutomi, M. (2009). A user-friendly method to geometrically calibrate projector-camera systems. In *Computer Vision and Pattern Recognition Workshops*, pages 47 – 54.

[Baker et al., 2011]Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M., and Szeliski, R. (2011). A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31.

[Bay et al., 2008]Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). Speeded-up robust features (surf). *Comput. Vis. Image Underst.*, 110(3):346–359.

[Björck, 1996]Björck, Å. (1996). *Numerical Methods for Least Squares Problems*. Siam.

[Bradley et al., 1991]Bradley, C., Vickers, G., and Tlusty, J. (1991). Automated rapid prototyping utilizing laser scanning and free-form machining. *CIRP Annals – Manufacturing Technology*, 41(1):437–440.

[Calonder et al., 2010]Calonder, M., Lepetit, V., Strecha, C., and Fua, P. (2010). Brief: Binary robust independent elementary features. In *Proceedings of the 11th European Conference on Computer Vision: Part IV*, pages 778–792.

[Fischler and Bolles, 1981]Fischler, M. and Bolles, R. (1981). RANdom SAmpling Consensus: a paradigm for model fitting with application to image analysis and automated cartography. *Commun. Assoc. Comp. Mach.*, 24:358–367.

[”Fitzgibbon et al., 1998]”Fitzgibbon, A. W., Cross, G., and Zisserman, A. (”1998”). ”automatic 3D model construction for turn-table sequences”. In *”3D Structure from Multiple Images of Large-Scale Environments, LNCS 1506”*, pages ”155–170”.

[Forssn and Lowe, 2007]Forssn, P.-E. and Lowe, D. G. (2007). Shape descriptors for maximally stable extremal regions. In *ICCV*. IEEE.

[Gauglitz et al., 2011]Gauglitz, S., Höllerer, T., and Turk, M. (2011). Evaluation of interest point detectors and feature descriptors for visual tracking. *International Journal of Computer Vision*, 94(3):335–360.

[Hartley and Sturm, 1997]Hartley, R. I. and Sturm, P. (1997). Triangulation. *Computer Vision and Image Understanding: CVIU*, 68(2):146–157.

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

[Jamil Draréni, 2009]Jamil Draréni, Sébastien Roy, P. S. (2009). Geometric video projector auto-calibration. In *Proceedings of the IEEE International Workshop on Projector-Camera Systems*, pages 39–46.

[Kazo and Hajder, 2012]Kazo, C. and Hajder, L. (2012). High-quality structured-light scanning of 3D objects using turntable. In *IEEE 3rd International Conference on Cognitive Infocommunications (CogInfoCom)* , pages 553–557.

[Lepetit et al., 2009]Lepetit, V., F.Moreno-Noguer, and P.Fua (2009). Epnp: An accurate o(n) solution to the pnp problem. *International Journal Computer Vision*, 81(2):155–166.

[Leutenegger et al., 2011]Leutenegger, S., Chli, M., and Siegwart, R. Y. (2011). Brisk: Binary robust invariant scalable keypoints. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, pages 2548–2555.

[Levi and Hassner, 2015]Levi, G. and Hassner, T. (2015). LATCH: learned arrangements of three patch codes. *CoRR*.

[Liao and Cai, 2008]Liao, J. and Cai, L. (2008). A calibration method for uncoupling projector and camera of a structured light system. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 770 – 774.

[Lowe, 1999a]Lowe, D. G. (1999a). Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision*, ICCV '99, pages 1150–1157.

[Lowe, 1999b]Lowe, D. G. (1999b). Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision*, ICCV '99, pages 1150–1157.

[Mair et al., 2010]Mair, E., Hager, G. D., Burschka, D., Suppa, M., and Hirzinger, G. (2010). Adaptive and generic corner detection based on the accelerated segment test. In *Proceedings of the 11th European Conference on Computer Vision: Part II*, pages 183–196.

[Martynov et al., 2011]Martynov, I., Kamarainen, J.-K., and Lensu, L. (2011). Projector calibration by "inverse camera calibration". In *SCIA*, volume 6688 of *Lecture Notes in Computer Science*, pages 536–544.

[Matas et al., 2002]Matas, J., Chum, O., Urban, M., and Pajdla, T. (2002). Robust wide baseline stereo from maximally stable extremal regions. In *Proc. BMVC*, pages 36.1–36.10.

[Moreno and Taubin, 2012]Moreno, D. and Taubin, G. (2012). Simple, accurate, and robust projector-camera calibration. In *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission, Zurich, Switzerland, October 13-15, 2012*, pages 464–471.

[Muja and Lowe, 2009]Muja, M. and Lowe, D. G. (2009). Fast approximate nearest neighbors with automatic algorithm configuration. In *In VISAPP International Conference on Computer Vision Theory and Applications*, pages 331–340.

[Nayar et al., 2006]Nayar, S. K., Krishnan, G., Grossberg, M. D., and Raskar, R. (2006). Fast separation of direct and global components of a scene using high frequency illumination. *ACM Trans. Graph.*, 25(3):935–944.

[Ortiz, 2012]Ortiz, R. (2012). Freak: Fast retina keypoint. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 510–517.

[Pablo Alcantarilla (Georgia Institute of Technolog), 2013]Pablo Alcantarilla (Georgia Institute of Technolog), Jesus Nuevo (TrueVision Solutions AU), A. B. (2013). Fast explicit diffusion for accelerated features in nonlinear scale spaces. In *Proceedings of the British Machine Vision Conference*. BMVA Press.

[Pal et al., 2012]Pal, C. J., Weinman, J. J., Tran, L. C., and Scharstein, D. (2012). On learning conditional random fields for stereo - exploring model structures and approximate inference. *International Journal of Computer Vision*, 99(3):319–337.

[Park and Park, 2010]Park, S.-Y. and Park, G. G. (2010). Active calibration of camera-projector systems based on planar homography. In *ICPR*, pages 320–323.

[Rosten and Drummond, 2005]Rosten, E. and Drummond, T. (2005). Fusing points and lines for high performance tracking. In *In Internation Conference on Computer Vision*, pages 1508–1515.

[Rublee et al., 2011]Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). Orb: An efficient alternative to sift or surf. In *International Conference on Computer Vision*.

[Sadlo et al., 2005]Sadlo, F., Weyrich, T., Peikert, R., and Gross, M. H. (2005). A practical structured light acquisition system for point-based geometry and texture. In *Symposium on Point Based Graphics, Stony Brook, NY, USA, 2005. Proceedings*, pages 89–98.

[Scharstein et al., 2014]Scharstein, D., Hirschmüller, H., Kitajima, Y., Krathwohl, G., Nesic, N., Wang, X., and Westling, P. (2014). High-resolution stereo datasets with subpixel-accurate ground truth. In *Pattern Recognition - 36th German Conference, GCPR 2014, Münster, Germany, September 2-5, 2014, Proceedings*, pages 31–42.

[Scharstein and Szeliski, 2002]Scharstein, D. and Szeliski, R. (2002). A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. *International Journal of Computer Vision*, 47:7–42.

[Scharstein and Szeliski, 2003]Scharstein, D. and Szeliski, R. (2003). High-accuracy stereo depth maps using structured light. In *CVPR (1)*, pages 195–202.

[Seitz et al., 2006]Seitz, S. M., Curless, B., Diebel, J., Scharstein, D., and Szeliski, R. (2006). A comparison and evaluation of multi-view stereo reconstruction algorithms. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006), 17-22 June 2006, New York, NY, USA*, pages 519–528.

[Sturm et al., 2012]Sturm, J., Engelhard, N., Endres, F., Burgard, W., and Cremers, D. (”2012”). ”a benchmark for the evaluation of rgb-d slam systems”. In *”Proc. of the International Conference on Intelligent Robot Systems (IROS)”*.

[Tola et al., 2010]Tola, E., Lepetit, V., and Fua, P. (2010). Daisy: An efficient dense descriptor applied to wide baseline stereo. *IEEE TRANS. PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 32(5).

[Tomasi, C. and Shi, J., 1994]Tomasi, C. and Shi, J. (1994). Good Features to Track. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 593–600.

[Xu and Aliaga, 2007]Xu, Y. and Aliaga, D. G. (2007). Robust pixel classification for 3d modeling with structured light. In *Proceedings of the Graphics Interface 2007 Conference, May 28-30, 2007, Montreal, Canada*, pages 233–240.

[Yamauchi et al., 2008]Yamauchi, K., Saito, H., and Sato, Y. (2008). Calibration of a structured light system by observing planar object from unknown viewpoints. In *19th International Conference on Pattern Recognition*, pages 1–4.

[Zhang, 2000]Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334.