Opto-Electronics Review 27 (2019) 202–212

Contents lists available at ScienceDirect

Opto-Electronics Review

journal homepage: http://www.journals.elsevier.com/opto-electronics review

Real-time camera pose estimation based on volleyball court view

K. Szelag*, P. Kurowski, P. Bolewicki, R. Sitnik

Faculty of Mechatronics, Warsaw University of Technology, 8 Św. Andrzeja Boboli St.,Warsaw 02-525, Poland

ARTICLE INFO

Article history:
Received 2 April 2019
Accepted 23 May 2019

Keywords:
Camera calibration
Pose etimation
Optical measurement
3d tracking
Volleyball
Pose estimation

ABSTRACT

The use of technology in sports has increased in recent years. One of the most influential of these technologies is referee support systems. Team sports such as volleyball require accurate and robust tracking systems that do not affect either the players or the court. This paper introduces the application of intrinsic and extrinsic camera calibration in a 12-camera volleyball referee system. Intrinsic parameters are calculated by using the classic pinhole model and Zhang's method. To perform extrinsic calibration in real time, the volleyball court is treated as a global calibration artifact. Calibration keypoints are defined as courtline intersections. In addition, a new keypoint detection algorithm is proposed. It enables achievement of an accurate camera pose in regard to the court. With all 12 cameras calibrated in a common coordinate system, a dynamic camera stereo pair creation is possible. Therefore, with known ball 2D image coordinates, the 3D real ball coordinates can be reconstructed and the ball trajectory can be estimated. The performance of the proposed method is tested on a synthetic data set, including 3Ds Max rendering and real data scenarios. The mean camera pose error calculated for data biased with keypoint detection errors is approximately equal to 0.013% of the measurement volume. For the real data experiment with a human hand phantom, it is possible to determine the presence of the human phantom on the basis of the ball reflection attitude.

1. Introduction

Camera pose estimation is an essential step in stereophotogrammetry and multi-camera systems [1]. Nowadays, owing to the decreasing cost of cameras, stereo camera systems are widely used in many applications in augmented reality, medicine [2,3], robotics [4], and sports [5]. However, use of dedicated markers in observed scenes is not always possible. The dynamic environment [6], mobile camera [7], and scene constraints are among many limitations. For example, in technologies connected with volleyball, it is forbidden to modify the ball, court, or net. Therefore, techniques that can estimate camera pose without observed scene modifications have been given increased attention in recent years. To determine the camera pose, the camera image coordinate system should be connected with real scenes being observed, e.g., with the known locations of characteristic objects and keypoint detection [8]. Hence, the real coordinate space should be connected with the image coordinate space. In recent years, the problem described above has been studied and some accurate solutions have been made possible. Currently, multicamera systems are becoming increasingly popular for sports refereeing assistance. Multi-camera systems are becoming increasingly popular for sports refereeing assistance. Beside the well-known Hawk-Eye system [5], which is used for estimation of the ball trajectory during tennis matches, there are many different systems that are used to support the referee, such as the goal lines [9] in football, or camera preview systems in volleyball [10]. These systems represent different approaches to aiding the referee. Hawk-Eye is a well-calibrated multi-camera system that can estimate the ball position in each frame set that is collected. However, system calibration is performed offline; that is, as both intrinsic calibration and pose estimation [11]. In that case, displacement of any camera due to, for example, the operator or a ball collision, may impede the system accuracy. In the volleyball challenge system, no image processing is employed. Referee support is based only on high (frame per-second (fps) replays. The most accurate and reliable results are given by the well-calibrated setup of the Hawk-Eye system. However, camera displacement in team sports played in a small court area (volleyball, basketball) is likely to occur. To address this issue, we propose real-time camera pose estimation. The proposed solution is a part of the OGX—BallTracking system, which implements a 2D ball trajectory tracking and a 3D coordinate reconstruction. In this paper, calibration, pose estimation and real-time re-calibration proce-

* Corresponding author.
E-mail addresses: k.szelag@mchtr.pw.edu.pl (K. Szelag), p.kurowski@mchtr.pw.edu.pl (P. Kurowski), p.bolewicki@mchtr.pw.edu.pl (P. Bolewicki), r.sitnik@mchtr.pw.edu.pl (R. Sitnik).

https://doi.org/10.1016/j.opelre.2019.05.005
1230-3402/© 2019 Association of Polish Electrical Engineers (SEP). Published by Elsevier B.V. All rights reserved.

dures for scenarios of camera displacement are introduced. Such displacement may occur on account of camera-ball, or camera-player collisions. Automatic real-time recalibration overcomes this problem. In addition, with real-time extrinsic calibration, the ball tracking system could be mobile and easy to situate and maintain.

2. Related work

2.1. Pose estimation methods: overview

To the estimate camera pose, an appropriate mathematical model should be constructed. Calibration of the camera in a monocular system with a single frame available requires well-defined 3D-2D model correspondence. The local and global approaches are used to achieve this goal. On the other hand, camera pose estimation could be performed in a well-known environment or in a natural partially unknown one, which requires another classification approach, specifically, detection and correlation approach. The groups of these approaches are discussed below.

2.1.1. Feature range

In local approaches, local, unique keypoints are detected both in camera-only [8] and laser-sensor-boosted [12] systems. The accuracy of local methods strongly depends on the quality of detected keypoints. Quality could be based on the number of keypoints (more keypoints result in higher quality results), an equal spatial distribution among them, and accurate detection possibility of points. The performances of these techniques are contingent on the stability of their environments with stable lighting [13]. Many feature descriptors of a scene being observed include Harris features [14], SIFT features [15], SURF features [5] and BRISK [16] features, which are represented by the descriptor vector and are calculated locally for the image sub region of interest (ROI) [17] (in 2D approaches) or in the k-nearest-neighborhood (for 3D approaches). Using known features, 2D-3D matching could be performed in the kd-tree Lowe approach [18], which is often used to obtain 2D-3D correspondences. With fewer points, the 3D point location known problem could be described as a perspective n point (pnp) problem, with many algorithmic solutions [19–22], which enable determining the object position [7]. The global feature approach uses statistical models of the observed scene combined with a well-known or online computed 3D model of the observed scene [23]. In general, global feature models are based on correlations with disparity minimization. Global features are well described by gradient distribution [24] or hierarchical models [25]. While not particularly useful in single frame setups, global models are widely used for motion detection, such as the optical flow technique [26]. For minimizing correlation errors and calculatingthe 2D-3D correspondence, Kalman filtering [27], Levenberg-Marquard optimization [28], or the RANSAC algorithm [29] may be used.

2.1.2. Known and unknown environment approaches

The amount of data available a priori knowledge is usually a key problem feature in 3D camera setups and calibration. With 3D artifacts or a set of 2D calibration artifacts, which represent fully available data, the traditional stereo calibration that is used in measurement systems may be performed [30,31]. In most cases, many features or environments are well determined but there is inadequate information to perform both intrinsic and extrinsic camera calibration. This could be overcome by offline intrinsic camera calibration. In papers focused on pose estimation, intrinsic camera parameters are often assumed to be known (e.g., [6]). Therefore, approaches using a priori known intrinsic camera parameters could not be deemed artifact-free methods. When describing unknown environment approaches and artifact-free methods, monoSLAM [32] is a state-of-the art method. Owing to the structure from the motion approach [33], which includes frame-to-frame correlation and calculation of both intrinsic and extrinsic camera parameters in a single optimization routine, monoSLAM can be used in unknown environments and uncalibrated camera setups.

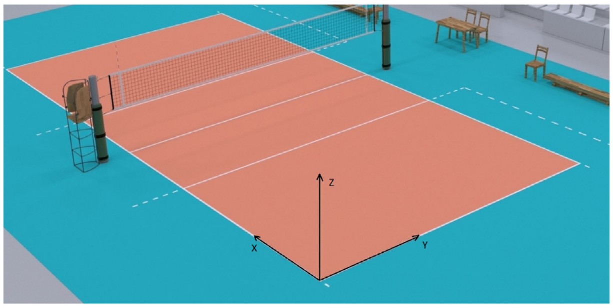

Fig. 1. Global coordinate space.

3. Proposed method

3.1. Mathematical model and coordinate system

3.1.1. Camera pinhole model

Among many different approaches, the pinhole camera model is one of the simplest and most robust and is, thus, chosen to describe camera behaviour. Therefore, the camera is represented by a set of lines intersecting in one point, the camera nodal point. Each line is also drawn via a single point representing the camera sensor pixel. In addition, let us define the local coordinate system with X and Y axes that are parallel to the camera sensor matrix, and the lens optical axis is z. In the defined coordinate space, the line equation for any point P(P_x, P_y) located in the camera sensor is:

$$\begin{cases} x = \dfrac{P_x}{f_x} \cdot t \\ y = \dfrac{P_y}{f_y} \cdot t \\ z = t \end{cases} \tag{1}$$

Let us define an additional coordinate system, the image coordinate space, scaled by the pixel width and height factor m and translated by vector v=[C_x,C_y,F]:

$$\{ \begin{matrix} u = P_x \cdot m - C_x \\ v = P_y \cdot m - C_y \end{matrix} \qquad <(2)$$

These two equations enable us to determine image pixel coordinates for point (X_l,Y_l,Z_1) described in local the camera coordinate system:

$$s \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} F_x & 0 & C_x \\ 0 & F_y & C_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} X_l \\ Y_l \\ Z_l \end{bmatrix} \tag{3}$$

where F_x=f_x·m, F_y=f_y·m

3.1.2. Global coordinate system

Let us define the global (real) coordinate system (Fig. 1).

3.2. Keypoints

3.2.1. Keypoint definition

In the beginning, let us define keypoints as points located in a 3D global coordinate system which could be located in a 2D image and are used for camera pose estimation. The proper choice of keypoints is one of the most important issues during successful camera

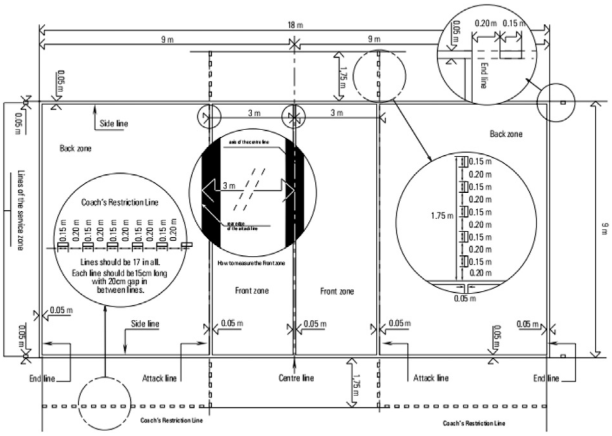

Fig. 2. Volleyball court dimensions [34].

calibration. When calibration artifacts are used, markers (planar or 3D) are used both as input for intrinsic calibration and as keypoints in extrinsic calibration. This approach is easier and more accurate than scene-feature-based approaches, but also not possible to be used in real-time calibration during a volleyball match. What we thus propose is to perform offline intrinsic calibration with calibration artifacts (described in the Intrinsic Calibration subsection) and to use the playing court as a natural artifact in extrinsic calibration. The sizes and locations of lines, nets and corners are well defined and normalized (Fig. 2). Especially, line intersections are considered to be well-defined and evenly distributed keypoints.

3.2.2. Keypoint detection

To detect line intersections in the observed scene, the detection algorithm with an adaptive ROI was developed. The adaptive ROI approach is widely used in complex scenes [35]. With a proper ROI assigned, it is possible to detect only court lines (with no artifacts) and to obtain line fragments, the length of which could be adapted to achieve better results in the following interpretation. In addition to known approaches [36–38], a priori information of line mutual placement and orientation for each corner is considered. The proposed algorithm is a modification of the Hough line Transform, which is widely used in computer vision [39]. The main steps of the algorithm are outlined as follows:

- Image preprocessing: image contrast increases and noise is filtered. Firstly, the image is converted to a grayscale. After conversion, histogram equalization is perfomed. After histogram equalization, Haar transformation [40] is applied. In Haar transformation, the subsequent noise reduction steps are performed. Then, the reverse Haar transformation is calculated. As a result, a grayscale image with a high contrast (a high-intensity difference between the object and background) is obtained (Fig. 3).
- Edge detection: After preprocessing, edges are detected with the Canny algorithm. Since the image is normalized after the previous step, a single set of Canny algorithm parameters could be used. Parameters are adjusted experimentally with a set of representative test frames. The Canny algorithm results was compared with the desired output. The best results are achieved with 0.4 threshold range and 0.8 of the pixel value range, as well as a 5 × 5 kernel.
- Hough transform and line set detection: let us define the line set (LS) associated with each keypoint as a set of lines detected after the edge detection step. For every line set, LS distribution (LD)

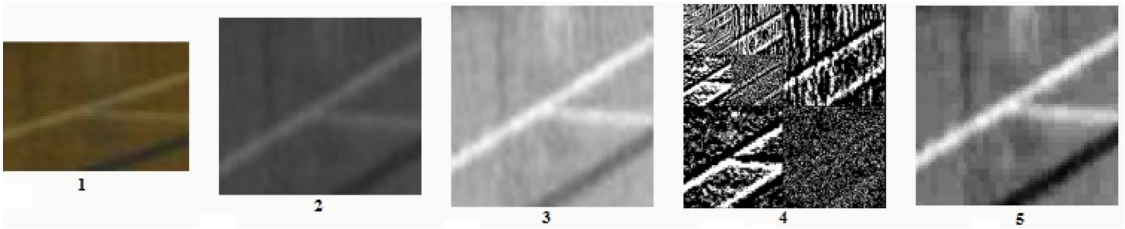

Fig. 3. Image preprocessing steps: 1- input image, 2- grayscale image, 3- histogram equalized, 4- Harr transformation, 5- output.

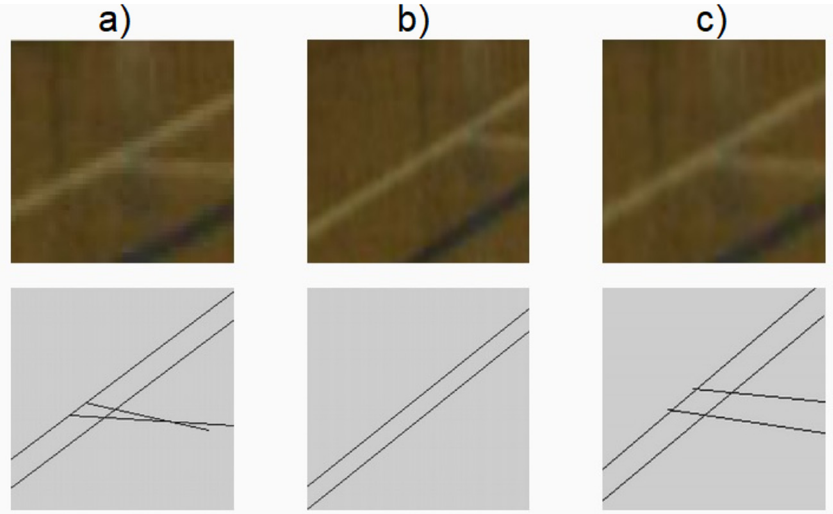

Fig. 4. Three steps of ROI adaptation: a) initial ROI for line detection and detected set of lines. The location and orientation of horizontal lines are improper - ROI is moved; b) mid-step ROI, no horizontal lines require an ROI width increase; c) final ROI - the proper line set is detected.

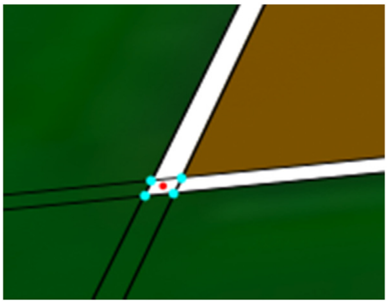

Fig. 5. Corner calculation idea.

can be defined as the number, placement, and orientation of the detected lines. Since volleyball court corners represent a finite number of possible LS shapes (T-shape Fig. 3, L-shape Fig. 5, etc.), the set of desired LD could be defined and the proper member of that set could be treated as line reference (LR) for the particular keypoint. Based on LD and LR comparison for each LS detected, it is possible to determine if there is not enough information in the defined ROI, or if there is too much noise. For both detecting LS and calculating LD from ROI, Hough transform [39] could be used. During transformation each pixel of the edge-filtered ROI votes for the line it could comprise and the value of the point with that line coordinates is incremented. After collecting "votes," the lines with a value higher than the threshold are considered detected LS. The threshold is experimentally adjusted and set to 0.6 of ROI greater dimensions. In order to compare LR and LD, a set of angle ranges is defined. The best experimental results are achieved with ranges distributed equally every $\pi/6$. Then, number of lines belonging to each range is counted and compared. The correlation of both sets (SD) is described by the sum of absolute differences between line numbers in each range. Based on SD, the LS Quality (LSQ) can be defined as the relative number of lines that are in correct ranges. Each time LS is detected, LD and LSQ are calculated. Then, if LSQ is greater than 0.9 (every line detected exists in the correct range and only some lines could be double detected). LS is hence considered adequate for further calculations.

- ROI localization prediction: Local region location and dimensions used in previous steps must be adjusted to achieve better results. Line density and distribution are compared with corner definitions. Then, transformation of ROI for the next iteration of the algorithm can be calculated. For example, if the detected line set contains too many horizontal lines and no vertical lines, the ROI location is moved horizontally, the horizontal dimension of ROI is decreased, and the vertical dimension is increased. Correction changes affect next-frame ROI in a way similar to a negative feedback control loop (Fig. 4). It is worth mentioning that the LSQ value is the target value in that step. The ROI adaptation algorithm is heuristic. During tests performed in this study, the "reset" condition was added: if LSQ continuously decreased, we changed the ROI to initial values. To maintain the real-time feature, the maximal number of iterations is defined. After that number, ROI is set to initial values and calculations are stopped for that frame.
- Corner detection: It is performed if and only if LSQ is adequate. Using the given set of four lines, four intersections are calculated. The keypoint is located in the center of mass of the quadrangle made by detected intersections (Fig. 5).

Exemplary results of corner detection in real time environment are shown in Fig. 6.

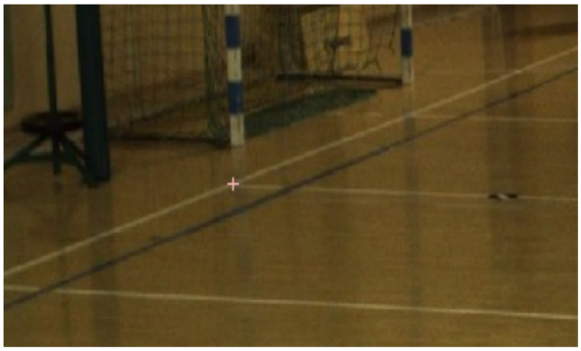

Fig. 6. Exemplary results of corner detection in a real environment.

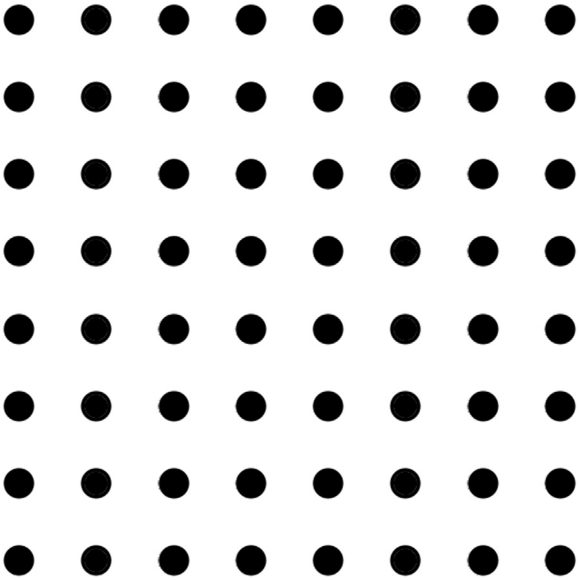

Fig. 7. Circular calibration pattern.

3.3. Calibration

As mentioned in the introduction, on account of a lack of keypoints detected in an observed scene, the calibration procedure must be divided into two parts. In the first, offline part, intrinsic calibration is obtained and distortion is corrected. In the second, online part, extrinsic calibration, the camera pose is calculated based on the keypoints detected.

3.3.1. Intrinsic calibration

The mathematical representation of scene components must be simple and easy to calculate owing to the problem properties, e.g., many cameras, few scene keypoints and real-time computation. Therefore, the pinhole camera model mentioned in the Introduction was chosen [41]. One of the main advantages of this model in the presented application is its global nature, which allows calibration results to be used in various depths. Therefore, a large measurement volume can be calibrated with a small calibration artifact. As a computational method of intrinsic parameters (focal length and center of projection), Zhang's method [42] was chosen. As a calibration artifact, a planar target pattern with 64 circular markers, spaced equally every 40 mm in both directions was used (Fig. 7).

Many solutions, especially in robotics, use a checkerboard pattern for geometric intrinsic calibration. However, the center of the circular marker could be accurately found with gradient based methods [43]. The intrinsic calibration routine is conducted as follows:

- Open the camera aperture and adjust the focal plane to a distance of 6.56 m. During a volleyball match, the camera is located on the side of the court 5.5 m above ground and at a distance of 5 m from the court line. Therefore, the distance between the camera lens and center of the measurement volume (middle of court is 1.25 m above the ground) is equal [Eq. (4)]:

$$d_c = \sqrt[2]{h^2 + d_e{}^3} \approx 6.56m \tag{4}$$

- Close the aperture to level when volleyball is visible with at least one third of maximum pixel intensity in every position of the measurement volume. This step is performed to achieve a maximal depth of field (from one external line to the second external line minimum). Cameras, lenses, and their parameters were chosen to fulfill these requirements.
- Gather calibration images 2 m × 2 m volume in front of the camera was equally divided into eight positions. For each position, five artifact images with different orientations were taken. Both the number of positions and number of images in each position were adjusted experimentally in order to obtain the best rms for calibration time relation. Better accuracy was achieved when using more artifact positions and orientations. Changes in calibration accuracy were irrelevant for more than 40 images. After the experiments, we concluded that 15 is an adequate number of images to achieve similar calibration accuracy (with a difference lower than 0.5%) to that achieved with 40 images. However, in the proposed method, owing to the offline properties of intrinsic calibration, the 40 images procedure is presented. It is also worth mentioning that the pinhole geometric procedure is possible with only two artifact positions, as described in Ref. 44.
- Calculate the camera parameters Zhang's procedure with radial and tangential distortion is used. Pattern markers are detected by our original method [45] and the optimization routine is taken from OpenCV library. Since camera lenses used in projected system have small distortion, no sophisticated distortion model [46,47] is required.

3.3.2. Camera pose model

When the camera is calibrated and proper parameters are calculated, it is possible to estimate the camera pose. With keypoints detected, as mentioned in 3.2.1, the problem could be described as a perspective-n-point problem in pinhole camera representation. Let us define matrix E as a transformation matrix between the camera local coordinate system and court global coordinate system [Eq. (5)]:

$$E = \begin{bmatrix} r_{11} & r_{12} & r_{13} & | & t_1 \\ r_{21} & r_{22} & r_{23} & | & t_2 \\ r_{31} & r_{32} & r_{33} & | & t_3 \end{bmatrix} \tag{5}$$

and projection matrix K which consist intrinsic parameters [Eq. (6)].

$$K = \begin{bmatrix} F_x & 0 & C_x \\ 0 & F_y & C_y \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

Coordinate system transformation could be presented [Eq. (7)].

$$\begin{bmatrix} X_l \\ Y_l \\ Z_l \end{bmatrix} = E \cdot \begin{bmatrix} X_g \\ Y_g \\ Z_g \end{bmatrix} \tag{7}$$

The complete equation of the perspective-n-point problem could be derived from Eq. (3) Eq. (5) and Eq. (6) [Eq. (8)]

$$s \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \cdot E \cdot \begin{bmatrix} X_g \\ Y_g \\ Z_g \end{bmatrix} \tag{8}$$

With the known matrix K calculated in the previous steps, there are 13 unknown parameters, which need to be calculated.

3.3.3. Pose calculation and resulting algorithm

The perspective-n-point problem is a typical algorithmic problem in stereovision. There are many solutions which depend on the problem properties: number of points, accuracy of point detection, point alignment, etc. It is worth mentioning that a linear analytic solution [48], P3p method [49], EPnp method [19], point set division [50], RANSAC [29] and iterative nonlinear approach [51] can be used. In the presented paper, the iterative optimization approach was used because it is relatively good, has aneven distribution of points in the measurement volume and a possible low accuracy of keypoint detection. Due to the online character of the problem being solved, an additive approach was chosen.

Let us define parameters used in algorithm description:

- Current transformation matrix E^i- Part of the projection equation [Eq. (8)], defined in Eq. (5) calculated in the current iteration.
- Stored transformation matrix E - last valid transformation matrix, algorithm output.
- Camera matrix K - Part of projection equation (3q. 3) defined in Eq. (6).
- Current camera target position $C^i{}_{xyz}$ - X,Y,Z position of camera in global (court) coordinate space calculated in current iteration.
- Camera target position C_{xyz} - last valid X,Y,Z position of camera in global (court) coordinate space.
- Dislocation distance C_d - Euclidean distance between $C^i{}_{xyz}$ and C_{xyz}
- Dislocation threshold D_t - experimentally adjusted, is used in order to filter out noise.
- Dislocation counter D_c - number of frames in which dislocation distance is greater than the dislocation threshold.
- Dislocation counter threshold D_{ct} - experimentally adjusted number of consecutive noise frames.

For each frame, the following steps are performed (Fig. 8):

- Estimate camera pose E^i as a perspective-n-point problem solution with known K [Fig. 8a)].
- Calculate camera target position $C^i{}_{xyz}$ from matrix E^i and stored camera target position C_{xyz} from matrix E. [Fig. 8b)].
- Calculate dislocation distance C_d. [Fig. 8c)].
- Check if C_d is lower than D_t. [Fig. 8d)].
- If the distance is lower than the threshold, update the stored camera pose E with a combination of two matrices calculated as follows: the translation is obtained as a mean of two transformation matrices. Rotation matrices are converted with Rodrigues transformation. Then, the mean value is computed and reverse transformation is performed [52]. [Fig. 8h)]. In addition, set the dislocation frame counter to 0. [Fig. 8i)].

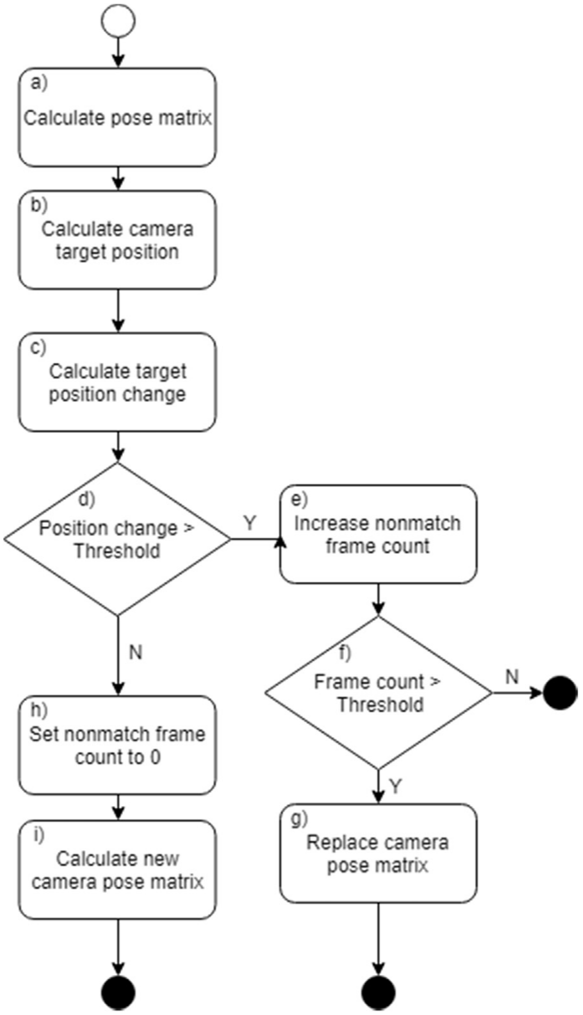
a)
Calculate pose matrix
b)
Calculate camera target position
c)
Calculate target position change
d)
Position change > Threshold
Y
e)
Increase nonmatch frame count
N
f)
Frame count > Threshold
N
h)
Set nonmatch frame count to 0
Y
i)
Calculate new camera pose matrix
g)
Replace camera pose matrix

Fig. 8. On-line camera pose adjustment algorithm.

- If D_d is greater than D_t the dislocation counter is incremented. [Fig. 8e)].
- If the dislocated frame count is greater than the threshold (D_{ct}) [Fig. 8f)], replace E with E_i. [Fig. 8g)].

3.4. Stereo system check

With all cameras calibrated it is possible to create stereo pairs and find keypoint 3D coordinates as pixel rays intersection. Then, distances between points and relative positions could be calculated and each camera calibration could be rated based on results achieved. Therefore, 3D re-projection inaccuracy could be used as additional pose estimation quality index.

4. Results and discussion

4.1. System setup

In proposed solution twelve, evenly placed stationary camera were used. As mentioned above, cameras were located on the sides of the court at a height of 5.5 m and a distance of 5 m from external line of the court. Cameras are oriented in a way, that upper generator of a projection cone is parallel to the volleyball court surface as shown in Fig. 9.

Distance of 5 m is a minimal distance consistent with official volleyball regulations [34]. Higher distance would decrease ball detection accuracy and, therefore, a 5 m distance was chosen. Cameras are located evenly: four cameras near each longer external line and two cameras near each shorter line as shown in Fig. 10. Due to the high speed of the volleyball, a high-speed, full-hd 180 Hz IDS UI-3060CP Rev.2 camera was used with 12,5 mm fixed focal length lenses. Typical exposures used during testing were between 0.7 and 1.5 ms. In order to achieve stereo system synchronization, we invented and built our own synchronizing system with optical-fiber connected and communicating camera synchronization modules.

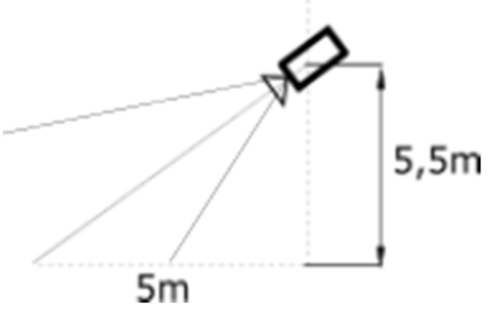

Fig. 9. Camera-court relative position and orientation.

4.2. Synthetic data

Each of algorithm steps was tested with synthetic data. Usage of synthetic data allows us to measure absolute error values without expensive use of specialized measurement devices, e.g. coordinate measuring machine [53]. 2D analysis steps (detection of circular markers and detection of keypoints) was also tested on real data in order to check influence of real environment noise and light inconsistency. As a real data test, reconstructed ball trajectory achieved with two cameras during volleyball training will be presented.

4.2.1. Keypoint detection

In order to prepare synthetic data for keypoint detection, 3Ds max environment with different camera poses was prepared. Camera positions are presented in Fig. 11. Camera pose could be described with three position coordinates and three camera target position coordinates. For each frame collected camera target is located in a volleyball court half center (Point A in Fig. 11). Camera coordinates' (X,Y,Z) distribution is described as follows:

- Y coordinate in global court coordinate system equals 5000 mm - it represents camera 5 m distant from longer external court line.
- Z coordinate takes three values: 5000 mm, 5500 mm, 6000 mm - representation of three height levels (length of camera mounting)
- X coordinate takes 40 values from 0 to 9000 - representation of evenly distributed camera location

With a coordinate change defined as above, 120 camera poses could be defined. Camera poses defined as above allow us to get corner images from various set of angles (in range expected in reality). For each camera pose a single frame was captured and keypoints were detected. As a reference, keypoints achieved from render data was considered.

Exemplary results of keypoint detection for single frame are presented in Fig. 12.

For each corner, difference between detected and reference point was calculated. For each coordinate histogram of coordinate error was calculated. Both histograms are presented in Fig. 13. Beside single coordinate error, euclidean distance in pixel coordinate space could be calculated. Histogram of errors described by euclidean pixel distance is presented in Fig. 14 Corner detection

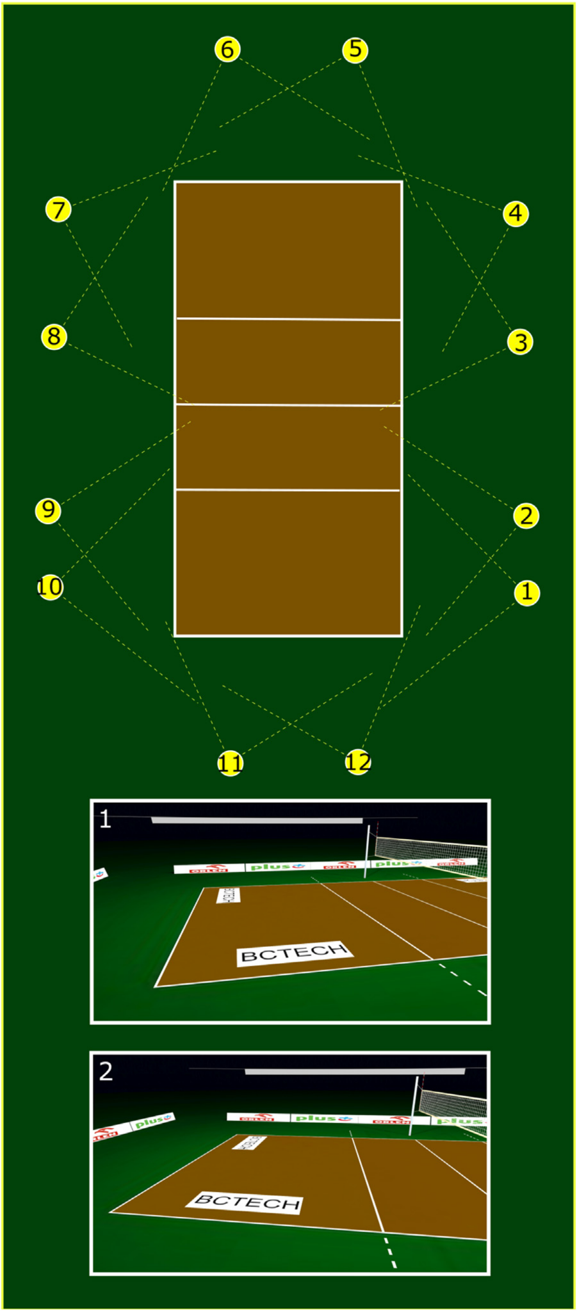

Fig. 10. Camera set distribution with projection cones. As an example views from camera 1 and camera 2 are presented.

Table 1
Corner detection error distribution parameters.

X (px)		Y(px)		Distance (px)	
Mean	stdDev	Mean	stdDev	Mean	stdDev
−0.011	0.226	−0.033	0.164	0.253	0.123

error could be estimated as a normal distribution. Mean values and standard deviation are gathered in Table 1.

4.2.2. *Pose estimation*

In order to estimate pose calculation accuracy, set of 8000 camera poses was rendered. Camera locations are defined as follows:

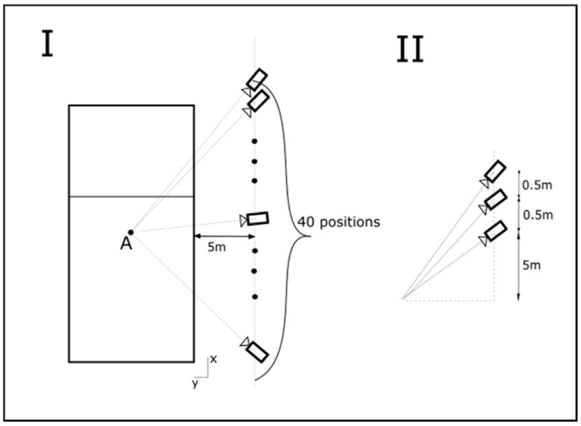

Fig. 11. Camera positions in synthetic data tests: I - plan view; A - center of court half; II - side view.

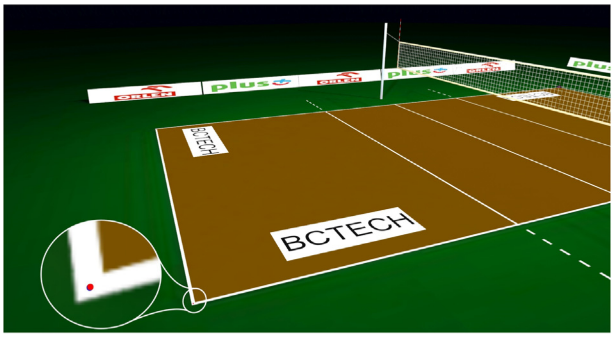

Fig. 12. Synthetic data corner detection. Red dot - detected corner coordinates, blue dot - reference corner coordinates.

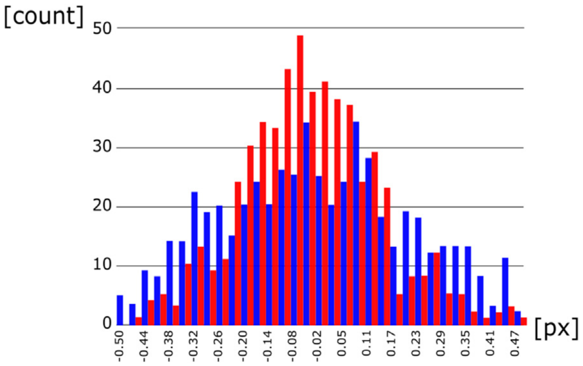

Fig. 13. Synthetic data corner detection error histogram. Blue - x coordinate difference [px], red - y coordinate difference [px].

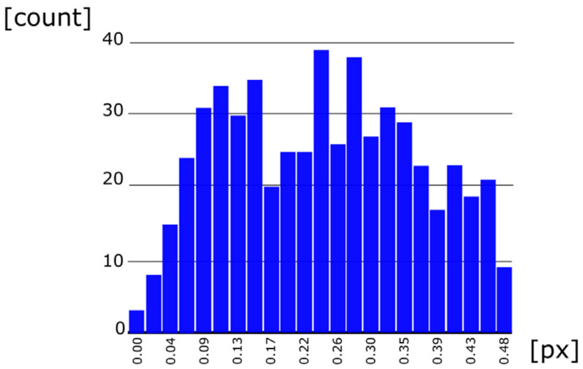

Fig. 14. Synthetic data corner detection distance error histogram. Blue - euclidean distance [px] values occurrences.

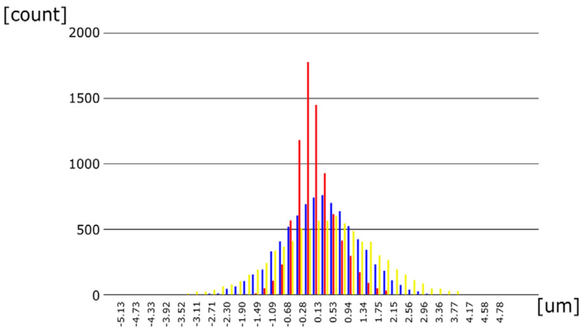

Fig. 15. Ideal synthetic data camera target pose error histograms. Red - x coordinate error [um], blue - y coordinate error [um], yellow - z coordinate error [um].

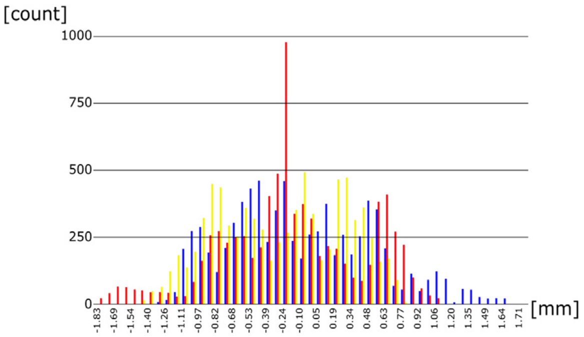

Fig. 16. Noised synthetic data camera target pose error histogram. Red - x coordinate error [mm], blue - y coordinate error [mm], yellow - z coordinate error [mm].

- x positions: starts form 1.5 m distant from shorter line of volleyball court, 20 positions every 0.2m
- y positions: starts from 5 m distant from longer line of volleyball court, 20 positions every 0.1 m
- z positions: starts from 5 m above ground, 20 positions every 0.1 m.

In each pose keypoint locations were calculated in two variants: with and without random noise added. Noise had a distribution similar to calculated for a keypoint detection in a prevous section (mean 0.25 px, stdDev 0.125). Camera matrix was calculated on the basis of render parameters. It was possible to calculate camera extrinsic parameters as described in section III.C.3. Additional point (0.0.6560) was defined in camera local coordinate system as a representation of camera target point (approximation of center of measurement volume). Distance 6.56 m was calculated in Eq. (4). In order to estimate both translation and rotation error, displacement of a target point instead of displacement of camera was considered. With the known rotation matrix R and the translation matrix T calculated camera target position was obtained with equation:

$$\begin{bmatrix} X_g \\ Y_g \\ Z_g \end{bmatrix} = R^T \cdot \begin{bmatrix} 0 \\ 0 \\ 6560 \end{bmatrix} - R^T \cdot T \tag{9}$$

After calculation of a camera position in global coordinate system, result value was compared with a reference camera position used in a render setup. In both ideal and noised cases, for each image, positioning x,y,z error was calculated and histogram was drawn (Figs. 15 and 16). The presented pose estimation error was calculated in a measurement volume of 9 m × 9 m × 3 m. Distance error mean equal to 1.03 mm represents to 0.08% of the measurement volume diameter (Tables 2 and 3).

Table 2
Camera pose calculation error for ideal case.

X (um)		Y(um)		Z (um)		Distance (um)	
Mean	stdDev	Mean	stdDev	Mean	stdDev	Mean	stdDev
0.013	0.6	−0.091	0.33	0.0697	0.796	0.935	0.492

Table 3
Camera pose calculation error for case with added noise.

X (mm)		Y(mm)		Z (mm)		Distance (mm)	
Mean	stdDev	Mean	stdDev	Mean	stdDev	Mean	stdDev
0.007	0.63	−0.16	0.61	0.26	0.55	1.03	0.343

4.2.3. Ball 3D coordinate reconstruction

In order to perform validation of the final system accuracy, synthetic data with two cameras and moving ball were created. The ball trajectory was designed as paraboloid with three rebounds, parallel to court longer line and with quasi-constant speed. As initial camera locations, camera 1 and 2 from Fig. 10 were taken. Therefore, cameras coordinates were equal to: (3 m,-5 m,z),(7.5 m,-5 m,z) in the court coordinates' system and were targeted at the center point of volleyball field half. Tests were performed for three heights: 5 m, 5.5 m and 6 m. For every camera location additional case with first camera moved by 0.1 m was considered. For each case sequence of ball positions were rendered, camera pose was calculated and 3D ball coordinates were reconstructed. During system validation with synthetic data two test were conducted:

1 Camera pose estimation without noise. During that test ball detection was contucted with (0.5;0.125) Gaussian noise. Reconstruction of 3D coordinates and error calculation with reference taken from render parameters was performed and error was calculated.
2 Pose estimation for two sets of cameras: initial (described in setup subsection) and moved by 0.1 m. Ball detection was constructed with (0.5; 0.125) Gaussian noise added. Reconstruction of 3D coordinates in both cases and disparity calculation was performed.

The first test allowed us to determine if the correct camera calibration resulted in accurate ball reconstruction coordinates even with ball detection noise added. The second test measured differences between results achieved with two slightly different camera positions. It determined if system accuracy will remain unchanged when random camera displacement occurs. For each camera pose case 3D coordinate reconstruction both trajectories and error histograms were similar. Exemplary trajectories and error histograms are shown in Fig. 17. Quantitative results are gathered and shown in Table 4.

4.2.4. Camera placement discussion

The camera placement error means that the value corresponds directly with the camera position. Since keypoints used in extrinsic calibration are detected on the camera image, better results are achieved when court corners are distributed more equally in the image captured. This occurs when the camera is located on a higher attitude. On the other hand, as presented in Table 4 a 3D ball reprojection error in particular dimension depends on the angle between the camera optical axis and that dimension directional vector. In projected system, the Z-axis error is very important due to system practical requirements (for example, to differ between hand-ball and court-ball reflections). Since the projected system has 12 cameras, error in X and Y directions could be easily compensated with data obtained from camera located near line orthogonal

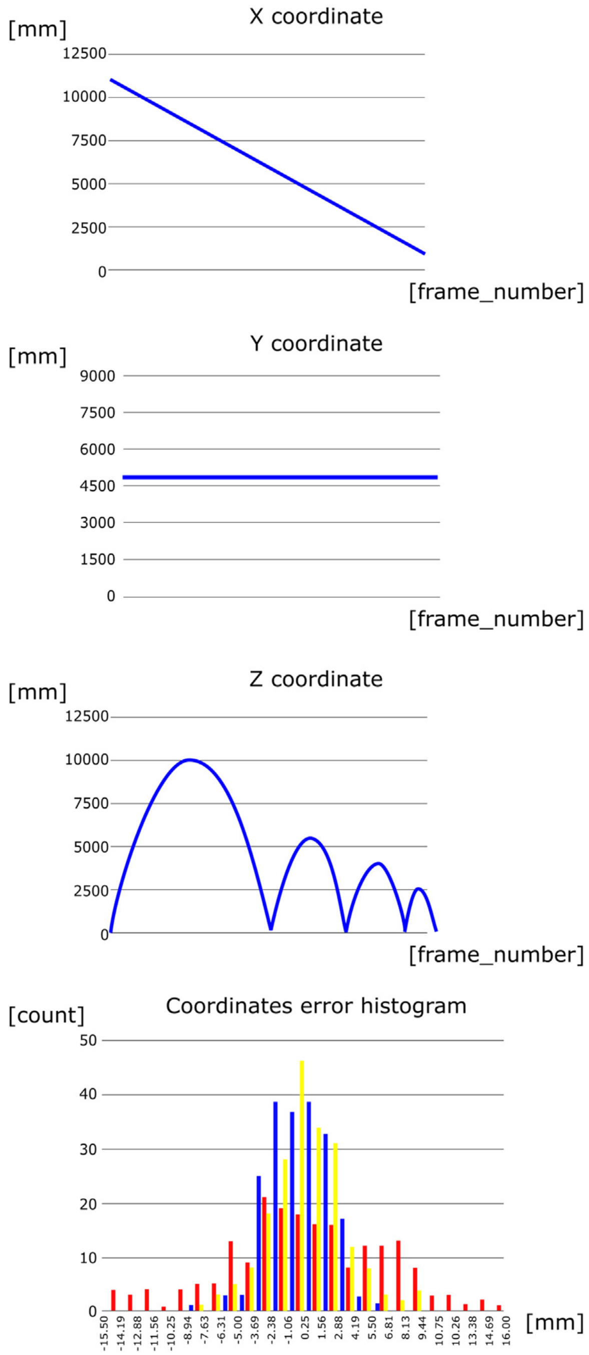

Fig. 17. Exemplary reconstructed ball trajectories and reconstruction error histograms.

to that direction. That is why camera positions described in section IV.A has been chosen as a compromise between pose calibration accuracy and 3D reconstruction of z coordinate.

4.3. *Real data*

In order to check system performance, a real-world environment experiment with usage of human hand phantoms and an automatic ball dispenser was conducted. The following scenarios were proposed:

1 Single reflection ball trajectory without human hand phantom
2 Single reflection ball trajectory with human hand phantom
3 Multi reflection ball trajectory without human hand phantom
4 Multi reflection ball trajectory with human hand phantom
5 Multi reflection ball trajectory with human hand phantom (first reflection on phantom)

Table 4
Quantitative results for ball trajectory 3D reconstruction error.

Height 5 m; not moved setup			
x_mean_error [mm]	−0.064	x_stdDev	2.327
y_mean_error [mm]	−0.008	y_stdDev	6.333
z_mean_error [mm]	0.102	z_stdDev	2.840
dist_mean_error [mm]	6.174	dist_stdDev	3.911
Height 5,5 m; not moved setup			
x_mean_error [mm]	−0.069	x_stdDev	2.380
y_mean_error [mm]	−0.012	y_stdDev	6.440
z_mean_error [mm]	0.110	z_stdDev	3.167
dist_mean_error [mm]	6.361	dist_stdDev	4.063
Height 6 m; not moved setup			
x_mean_error [mm]	−0.073	x_stdDev	2.437
y_mean_error [mm]	−0.015	y_stdDev	6.560
z_mean_error [mm]	0.119	z_stdDev	3.519
dist_mean_error [mm]	6.575	dist_stdDev	4.235
Height 5 m; moved setup			
x_mean_error [mm]	−0.062	x_stdDev	2.3
y_mean_error [mm]	−0.003	y_stdDev	6.216
z_mean_error [mm]	0.102	z_stdDev	2.801
dist_mean_error [mm]	6.08	dist_stdDev	3.823
Height 5,5 m; moved setup			
x_mean_error [mm]	−0.067	x_stdDev	2.351
y_mean_error [mm]	−0.008	y_stdDev	6.32
z_mean_error [mm]	0.109	z_stdDev	3.12
dist_mean_error [mm]	6.266	dist_stdDev	3.973
Height 6 m; moved setup			
x_mean_error [mm]	−0.07	x_stdDev	2.409
y_mean_error [mm]	−0.012	y_stdDev	6.437
z_mean_error [mm]	0.118	z_stdDev	3.464
dist_mean_error [mm]	6.474	dist_stdDev	4.139

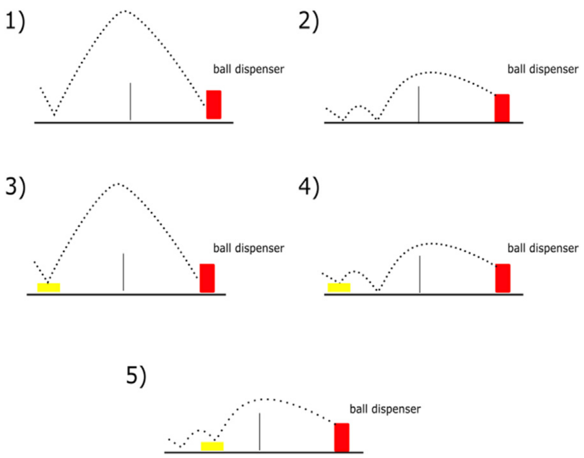

Fig. 18. Human hand phantom test scenarios.

Described scenarios are presented in Fig. 18. In each scenario of a full ball trajectory was reconstructed and the rebound attitude was calculated. Experimental results are presented in the calculated attitude histograms in Fig. 19. For cases without human hand phantoms, the mean attitude calculated was equal to 113.33 mm and standard deviation equal to 3.8 mm. For situations with human phantom mean attitude was equal to 133.62 mm and standard deviation equal to 3.8 mm. The phantom tests were performed in order to determine if accuracy of system is good enough to detect human hand between ball and court during match. As shown in Fig. 19,

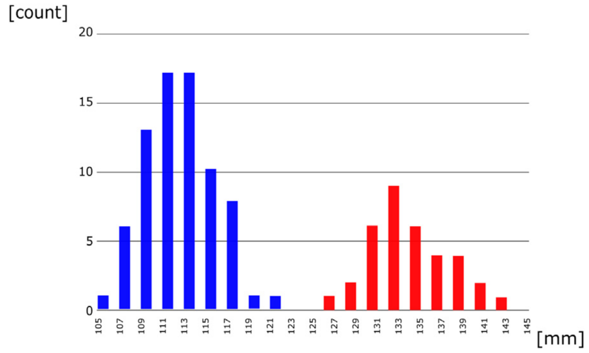

Fig. 19. Real data ball rebound with and without phantom calculated attitudes. Blue - attitude without phantom, red - attitude with phantom.

distributions of attitude values allow us to develop a recognition algorithm, which will be used as a module in referee aiding system.

5. Conclusion and futore research directions

The system presented in this paper enables measurement of the volleyball ball position 180 times per second with an accuracy equal to 2–4 mm. It has the ability to determine its components' positions and to refine its own calibration. That feature is important when use in sports is considered. Performed tests showed that system accuracy is good enough to support referee decisions and present reconstructed ball trajectories. With cameras calibrated intrinsically, the system configuration in new courts is fast and possible to be conducted by semi-professional technicians. The ability of real-time calibration, as mentioned in the Introduction, allows us to overcome displacement problems and makes the system even more mobile and easy to maintain. The proposed method is designed for camera pose estimation in volleyball courts, but it can also be used in multicamera systems operating in well-known environments, such as: surveillance, medicine, television studios, etc. Achieved accuracy of 0.01% in simulation of a measurement volume is considered to be good enough for realtime applications mentioned above. Tests performed in real environment show that system is able to detect collision with human phantom during typical use-case of the proposed system. There are still many fields where performance of system could be upgraded. For example, camera relative position detection could decrease influence of volleyball court dimensions' accuracy (line width, corner locations, etc.). With nonzero camera pose estimation error each of stereo pair will produce a slightly different 3D trajectory for a ball being detected. Destined system contains 12 cameras which are paired in 66 stereo set-ups. In order to get more reliable referee decisions based on that data, additional calibration step for whole set of cameras and stereo pairs has to be designed. Cameras are often visible in another camera field of view. This information could be used to determine relative position between cameras - especially camera movement (determine which camera has been moved). While analyzing system tests, we claim that there is still not enough data from real volleyball match, which will allow us to determine system final practical quality required in real-life system implementation. At this point data is being collected, labeled and prepared for future usage in the system evaluation. Also additional tests including real-life volleyball match scenarios have to be performed. It will ensure us that the designed system has desired robustness and will validate conclusions taken in this paper.

Acnkowledgments

We would like to thank Wojciech Zauski for 3D scene model creation and assistance with synthetic data generation.

References

[1] J.M. Frahm, K. Köser, R. Koch, Pose estimation for multi-camera systems, in: Joint Pattern Recognition Symposium, Springer, Berlin, Heidelberg, 2004, 286–293.
[2] K.W. Nam, J. Park, I.Y. Kim, K.G. Kim, Application of stereo-imaging technology to medical field, J. Healthc. Inf. Res. 18 (3) (2012) 158–163.
[3] M. Witkowski, R. Sitnik, M. Kujawińska, W. Rapp, M. Kowalski, B. Haex, S. Mooshake, 4D measurement system for automatic location of anatomical structures Biophotonics and New Therapy Frontiers, 6191, SPIE, 2006, 61910H.
[4] M. Achtelik, T. Zhang, K. Kuhnlenz, M. Buss, Visual tracking and control of a quadcopter using a stereo camera system and inertial sensors, IEEE Int. Conf. Robot Autom., 2009.
[5] B. Bal, G. Dureja, Hawk eye: a logical innovative technology use in sports for effective decision making, Sport Sci. Rev. 21 (1-2) (2012) 107–119.
[6] G. Bleser, H. Wuest, D. Stricker, Online camera pose estimation in partially known and dynamic scenes, in: IEEE/ACM ISMAR, 2006, 56–65.
[7] S. Ohayon, E. Rivlin, Robust 3d head tracking using camera pose estimation ICPR, 1, IEEE, 2006, 1063–1066.
[8] V. Lepetit, P. Lagger, P. Fua, Randomized Trees for Real-Time Keypoint Recognition, IEEE 2, 2007, 775–781.
[9] Wikipedia Page, Goal Line Technology, 2017.
[10] V-challenge Tds International System Webpage, 2019 (Accessed 20 February 2018) http://www.tdsinternational.eu/about-us.html.
[11] P. Tan, Y. Li, Z.Y. Huang, Feasibility analysis of "Hawkeye" technique employed in volleyball competition, J. Mianyang Normal Univ. 8 (2010) 028.
[12] D. Scaramuzza, A. Harati, R. Siegwart, Extrinsic self calibration of a camera and a 3d laser range finder from natural scenes, IEEE/RSJ IROS, 2007, 4164–4169.
[13] B. Triggs, Detecting Keypoints with Stable Position, Orientation, and Scale under Illumination Changes. ECCV, Springer, Berlin, Heidelberg, 2004, 100–113.
[14] C.G. Harris, M. Stephens, A combined corner and edge detector, AVC 15 (50) (1988) 10–5244.
[15] H. Zhou, Y. Yuan, C. Shi, Object tracking using SIFT features and mean shift, CVIU 113 (3) (2009) 345–352.
[16] S. Leutenegger, M. Chli, R. Siegwart, BRISK: Binary Robust Invariant Scalable Keypoints, IEEE ICCV, 2011, 2548–2555.
[17] R. Brinkmann, The Art and Science of Digital Compositing: Techniques for Visual Effects, Animation and Motion Graphics, Morgan Kaufmann, 2008.
[18] D.G. Lowe, Distinctive image features from scale-invariant keypoints, IJCV 60 (2) (2004) 91–110.
[19] V. Lepetit, F. Moreno-Noguer, P. Fua, EPnP: efficient perspective-n-point camera pose estimation, IJCV 81 (2009) 155–166.
[20] F. Moreno-Noguer, V. Lepetit, P. Fua, Accurate Non-Iterative O (n) Solution to the Pnp Problem, IEEE ICCV, 2007, 1–8.
[21] C.P. Lu, G.D. Hager, E. Mjolsness, Fast and globally convergent pose estimation from video images, IEEE Trans. Pattern Anal. Mach. Intell. 22 (6) (2000) 610–622.
[22] L. Quan, Z. Lan, Linear n-point camera pose determination IEEE, Trans. Pattern Anal. Mach. Intell. 21 (8) (1999) 774–780.
[23] V. Sharma, P.C. Barnum (2016). U.S. Patent No. 9,237,340. Washington, DC: U.S. Patent and Trademark Office.
[24] D.L. Ruderman, W. Bialek, Statistics of natural images: scaling in the woods, Adv. Neural Inf. Process. Syst. (1994) 551–558.
[25] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, IEEE Trans. Pattern Anal. Mach. Intell. 4 (2002) 509–522.
[26] B.K. Horn, B.G. Schunck, Determining optical flow, Artif. Intell. 17 (1-3) (1981) 185–203.
[27] A. Litvin, J. Konrad, W.C. Karl, Probabilistic video stabilization using Kalman filtering and mosaicing, Int. Society for Optics and Photonics 5022 (2003) 663–675.
[28] J.J. Moré, The Levenberg-marquardt Algorithm: Implementation and Theory. Numer. Anal, Springer, Berlin, Heidelberg, 1978, 105–116.
[29] M.A. Fischler, R.C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Commun. ACM 24 (6) (1981) 381–395.
[30] D. Svedberg, S. Carlsson, Calibration, pose and novel views from single images of constrained scenes, Pattern. Recognit. Lett. 21 (13-14) (2000) 1125–1133.
[31] C. Colombo, D. Comanducci, A. Del Bimbo, Camera Calibration with Two Arbitrary Coaxial Circles. ECCV, Springer, Berlin, Heidelberg, 2006, 265–276.
[32] A.J. Davison, I.D. Reid, N.D. Molton, O. Stasse, MonoSLAM: real-time single camera SLAM, IEEE Trans. Pattern Anal. Mach. Intell. 6 (2007) 1052–1067.
[33] C. Tomasi, T. Kanade, Shape and motion from image streams under orthography: a factorization method, IJCV 9 (2) (1992) 137–154.
[34] Official Fivb Rules for Volleyball, 2019, Accessed 20 February 2018 http://www.fivb.org/EN/Refereeing-Rules/documents/FIVB-Volleyball_Rules2013-EN_20121214.pdf.
[35] D. Ding, C. Lee, K.Y. Lee, An Adaptive Road ROI Determination Algorithm for Lane Detection (TENCON 2013), IEEE, 2013, 1–4.
[36] L.A. Fernandes, M.M. Oliveira, Real-time line detection through an improved Hough transform voting scheme, Pattern Recogn. 41 (1) (2008) 299–314.
[37] E. Rosten, T. Drummond, Machine Learning for High-speed Corner Detection. ECCV, Springer, Berlin, Heidelberg, 2006, 430–443.

[38] F. Mokhtarian, R. Suomela, Curvature scale space for robust image corner detection, Proc. ICPR (Cat. No. 98EX170) 2 (1998) 1819–1821.
[39] J. Illingworth, J. Kittler, A survey of the Hough transform, Comput. Gr. Image Process. 44 (1) (1988) 87–116.
[40] A. Chambolle, R.A. De Vore, N.Y. Lee, B.J. Lucier, Nonlinear wavelet image processing: variational problems, compression, and noise removal through wavelet shrinkage, IEEE Trans. Image Process. 7 (3) (1998) 319–335.
[41] M. Axholt, Pinhole Camera Calibration in the Presence of Human Noise (Doctoral Dissertation), Linköping University Electronic Press, 2011.
[42] Z. Zhang, A flexible new technique for camera calibration, IEEE Trans. Pattern Anal. Mach. Intell. (2000) 22.
[43] J. Heikkila, Geometric camera calibration using circular control points, IEEE Trans. Pattern Anal. Mach. Intell. 22 (10) (2000) 1066–1077.
[44] K. Szelag, G. Maczkowski, R. Gierwialo, A. Gebarska, R. Sitnik, Robust geometric, phase and colour structured light projection system calibration, Opto-Electron Rev. 25 (4) (2017) 326–336.
[45] R. Sitnik, M. Kujawinska, W. Załuski, 3DMADMAC system: optical 3D shape acquisition and processing path for VR applications, Optical Methods for Arts and Archaeology, SPIE 5857 (2005) 58570E.
[46] X. Mei, S. Yang, J. Rong, X. Ying, S. Huang, H. Zha, Radial Lens Distortion Correction using Cascaded One-parameter Division Model. ICIP, IEEE, 2015, pp. 3615–3619.
[47] Y. Tang, Y. Li, J. Luo, Parametric distortion-adaptive neighborhood for omnidirectional camera, Appl. Opt. 54 (23) (2015) 6969–6978.
[48] Y.I. Abdel-Aziz, H.M. Karara, M. Hauck, Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry, Photogramm. Eng. Remote Sens. 81 (2) (2015) 103–107.
[49] X.S. Gao, X.R. Hou, J. Tang, H.F. Cheng, Complete solution classification for the perspective-three-point problem, IEEE Trans. Pattern Anal. Mach. Intell. 25 (8) (2003) 930–943.
[50] S. Li, C. Xu, M. Xie, A robust O (n) solution to the perspective-n-point problem, IEEE Trans. Pattern Anal. Mach. Intell. 34 (7) (2012) 1444–1450.
[51] D.W. Marquardt, An algorithm for least-squares estimation of nonlinear parameters, J. Soc. Ind. Appl. Math. 11 (2) (1963) 431–441.
[52] R. Hartley, J. Trumpf, Y. Dai, H. Li, Rotation averaging, IJCV 103 (3) (2013) 267–305.
[53] Wikipedia page, Coordinate Measuring Machine, 2017.