©2021 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes,creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. DOI: 10.1109/ICCE50685.2021.9427685

Measuring vertical jump height using a smartphone camera with simultaneous gravity-based calibration

Fritz Webering[1], Leo Seeger[2], Niklas Rother[1] and Holger Blume[1]

***Abstract*—Vertical jump height is an important tool to measure athletes' lower body power in sports science and medicine. Several different methods exist to measure jump height, but each has its own limitations. This work proposes a novel way to measure jump height directly, using optical tracking with a single smartphone camera. A parabolic fall trajectory is obtained from this video by tracking a single feature. The parabolic trajectory is then used to partially calibrate the camera and convert pixel measurements into real-world units, allowing the calculation of the achieved height. Comparison to an optical motion capture system yields promising results.**

Keywords: vertical jump height, sports, camera calibration, gravity, parabola

I. INTRODUCTION

The assessment of vertical jump height is an important tool in sports sciences and sports medicine, used to assess ballistic lower body strength and power output. Vertical jump height, as defined by Bobbert and van Ingen [1] is the maximum vertical movement of the body center of mass. However, determining the exact center of mass is an involved procedure, because it depends on the movement of all body parts, including flexible tissue. The most accurate methods perform full-body motion capture of the jump and then determine the center of mass over time by summing the torques of all body segments as described by Aragn-Vargas [2].

Because of this difficulty, most practical jump height measurement methods determine the height indirectly from a different measured quantity, such as:

- Integrating launching force over time using force plates.
- Mechanically determining the highest point reached with an outstretched arm.
- Measuring flight time using a floor contact detection system.
- Measuring flight time through the analysis of high speed video. [3].
- Measuring flight time using inertial measurement units (IMUs).

The method proposed in this work is most similar to the motion capture method, in that it directly calculates the jump height as the difference between the highest point and the standing height, without measuring flight time. However, unlike a motion capture system, which is typically very expensive, our method can be performed using a simple smartphone camera, which is widely available.

*This work was not supported by any organization

[1]{webering,rother,blume}@ims.uni-hannover.de Leibniz University Hannover, Institute of Microelectronic Systems, Appelstr. 4, 30167 Hannover, Germany

[2]leo.seeger@stud.uni-hannover.de

II. RELATED WORK

We are not aware of any other published method to calculate jump heights by measuring distances in a video, while simultaneously using the jump trajectory itself to calibrate the system. The work by Balsalobre-Fern et al. [3] also uses high speed video to measure jump height, but calculates the flight time by counting the number of frames between launch and landing, not by measuring image distances.

The idea to calibrate cameras using the parabolic trajectories of objects affected by gravity has been researched previously by several groups. Sturm and Quan [4] propose the use of of parabolic trajectories of launched objects to estimate the infinite homography and some of the intrinsic parameters as well as camera pose. Another approach proposed by Zhang [5], Wu et al. [6] and Qi et al. [7], [8] uses a rigid one-dimensional object, such as a thrown stick to estimate the camera intrinsics.

An approach to perform a full camera calibration including pose estimation is presented by Chen et al. in [9], who use multiple parabolic trajectories of a bouncing ball and the known frame rate to calculate the intrinsic and extrinsic camera parameters of multiple cameras.

These approaches require multiple observations and provide very limited accuracy when only one parabolic trajectory is used for calibration [9]. However, we can reduce the degrees of freedom from 6—for a camera pose estimation—to 1, since we are only interested in a linear distance metric, as discussed in Section IV. This simplifies the calibration procedure and makes it more robust when only a single parabolic trajectory is used.

III. EXPERIMENTAL SETUP

In order to acquire test data to verify the algorithm, we conducted a small study with 6 healthy male participants, aged 25-35. All participants were briefed about the test procedure and provided informed consent.

Each participant was asked to perform a series of roughly ten counter movement jumps, hands resting on the hip, with a pause of two seconds after each jump. Additionally, the participants were asked to keep their gaze straight ahead while jumping, and not move their head up or down, in order to minimize relative movement between face and body center of mass. For subject no. 6, only 5 usable measurements exist, because a smartphone battery failed during the test.

For all trial jumps, the subject was outfitted with two retro-reflective motion capture markers and two 2D-codes optimized for computer vision. For the 2D-code markers we selected ArUco markers [10], [11], because of the ease of use and the quick and reliable detection. One marker of each type was

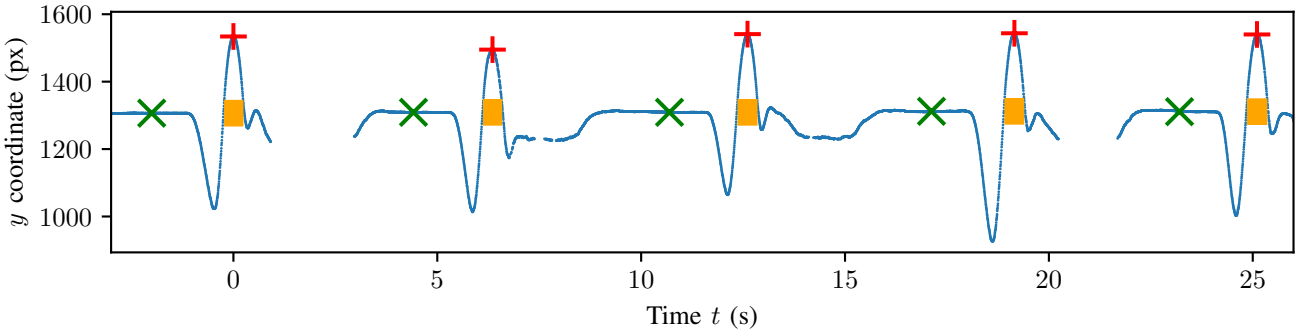

Fig. 1. Trajectory of the first 5 jumps of one subject, as tracked by the face detection algorithm. The × represents the standing height, the + marks the highest point of the jump, and the bar ▮ marks the length of the detected free-fall phase. Note the gaps in the plot where no face was detected, and the dip in the standing height (e.g. at $t = 15\,\mathrm{s}$) where the subject looked down while placing their feet back on the marked spot.

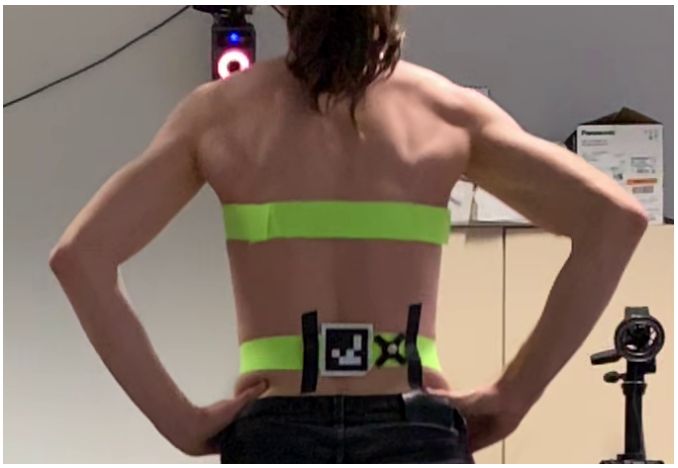

Fig. 2. Marker configuration in the lower back region, attached to an elastic strap. Next to the ArUco marker, the motion capture marker can be seen on an X-shaped touch fastener. An equivalent marker assembly was attached to the chest of the subject using the second strap.

attached to an elastic strap. Of the two marker sets, one was attached to the chest of the subject and the other to the lower back above the sacrum, as shown in figure 2.

In order to obtain reference height measurements, each trial was recorded with an 8-camera optical motion capture system from Vicon, yielding a three-dimensional trajectory for each of the two markers. A full calibration of the system was performed before the test run and the location error reported by the calibration software was below $1\,\mathrm{mm}$.

As suggested by Aragn-Vargas [2], tracking the movement of the body center of mass using an optical motion capture system is a suitable criterion reference (or 'gold standard') for jump height measurements. For this preliminary trial we opted to use only a single marker, attached near the center of mass, to calculate the reference heights. We argue that this is a close enough approximation of the real center of mass, since the subject's extremities do not move significantly during jump: The legs were kept straight, the arms were resting on the hip, and the head was kept level during the whole jump. To test this assumption, we applied the parabola fitting method (Section V-E) to the motion capture trajectory and were able to verify that the sacrum marker indeed closely resembles a parabolic trajectory with second-order coefficient $a = -\frac{g}{2}$. Leard et al. [12] used the same procedure and obtained good correlation with other methods, even though the subjects in that test did not keep their arms static during the jump.

Additionally, each trial run was recorded with two smartphones placed on a tripod in portrait orientation. The smartphone in front of the subject (year 2015 model) recorded 240 frames per second (FPS) with a resolution of $1280\times720\,\mathrm{px}^2$. The smartphone behind the subject (year 2018 model) recorded 240 FPS with a resolution of $1920\times1080\,\mathrm{px}^2$. The high frame-rates were chosen to minimize the exposure time, because our pre-tests showed greatly increased motion blur when recording jumps at 60 fps with the default camera settings.

In order to align the image plane parallel to the direction of gravity, as explained in Section IV, we used the app phyphox [13] to determine the orientation of the smartphone using the built-in accelerometer. This approach assumes that the accelerometer is calibrated in such a way, that its vertical axis is parallel to the image sensor.

IV. Camera Model

For the camera, we use the standard pinhole model. In order to correct the lens distortion one would normally perform a camera calibration. However, it appears that smartphone manufacturers already correct the lens distortion before the video is recorded. We performed checkerboard calibrations with the used smartphone cameras, but found no noticeable improvement compared to the uncorrected images. Thus, we used the unmodified videos for the evaluation.

The perspective projection of a pinhole camera can be described by the following equation:

$$\omega \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} a & 0 & b & 0 \\ 0 & c & d & 0 \\ 0 & 0 & e & f \\ 0 & 0 & -1 & 0 \end{pmatrix} \cdot \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix}, \tag{1}$$

where $a \dots f$ are constants, $(X_c, Y_c, Z_c, 1)$ is a point in the camera coordinate system, in which the XY plane is parallel to the image sensor. On the left side, (x, y) is a projected point in image coordinates, with depth z, and a scale factor ω.

From this equation, it is apparent that y depends only on Y_c and Z_c. As long as Z_c is constant, the transformation from Y_c to y is linear. Thus, any vertical movement along the Y axis of the camera coordinate system will translate to a change of the y image coordinate only, without any perspective distortion.

Since we used the built-in accelerometer to align the Y axis of the camera parallel to the axis of gravity, any purely vertical motion will be be mapped linearly to image y coordinates. The constant distance Z_c only determines the relative scale of the movement. This property used is in Section V-E to justify the validity of the calibration procedure.

V. ALGORITHM

The proposed algorithm for determining jump height consists of the following steps, which will be elaborated further in the following subsections:

(V-A) First, it is necessary to obtain the trajectory of a point on the subject's body.
(V-B) Starting from such a trajectory, it is necessary to identify the individual jumps.
(V-C) For each peak, the standing height before the jump is calculated by looking for a phase with little movement.
(V-D) After that, the parabolic free-fall trajectory can be extracted for each jump.
(V-E) A parabola is fitted to these data points in order to calibrate the absolute scale of the jump.
(V-F) Finally, the jump height in real-world units can be calculated using the standing height, the vertex of the parabola, and the calibration coefficient.

A. Marker Tracking

First, we have to obtain from a video the trajectory of a point on the subject's body, preferably a point on a firm surface close to the body center of mass [1]. Tracking other points further away from the center of mass—like the face—is also possible. However, this is likely to reduce the accuracy of the height measurement, because the face does not follow a true ballistic parabola, but is offset by the movement of the neck and the vertebral column.

In order to obtain a trajectory from the recorded video, any suitable object detection and tracking framework can be used. In this study we tracked two types of features: the subject's face and dedicated tracking markers.

Marker tracking was implemented using the OpenCV ArUco marker library [10], [11]. Face tracking was implemented using a pre-trained Caffe-based residual deep neural network that ships with OpenCV 3.4 [1]. Both use the center of the detected bounding box as the tracking point. An example face tracking trajectory $y_{\text{face}}(t)$ is shown in Figure 1.

[1] `res10_300x300_ssd_iter_140000.caffemodel`

B. Peak Detection

Since the trajectory likely contains small peaks like a bouncing motion after landing, it is necessary to devise a method to isolate only the actual jumps. We investigated multiple avenues, like limiting the absolute peak height to a specific range, requiring a minimum distance between peaks, and imposing a minimum peak width. The most robust method we found, was selecting only peaks with a topographic prominence of at least 0.4φ, where φ is the peak-to-peak amplitude of the whole trajectory.

C. Standing Height

Our test protocol included a quiescent phase of at least two seconds before each jump, where the subject moves as little as possible. The purpose of this phase is to determine the standing height, from which the jump height is measured.

Starting from each peak to the left, a window search was employed to find the quiescent height h_0 before that peak, which is required by our test protocol. The window search looks for an interval of length $1\,\text{s}$, where the peak-to-peak height difference inside that interval is less than 0.02φ.

D. Parabola Extraction

After that, it is necessary to determine the free fall phase of the jump, where the subject is not in contact with the ground. Because the feet are extended during the launch phase and are still in contact with the ground, it is not sufficient to simply select the part of the trajectory above the standing height.

In order to account for the length of the feet, we selected the part of the trajectory where $y > h_0 + 0.1\varphi$. Only the portion of the trajectory which satisfies this condition is used for the following steps. Two example free-fall trajectories are shown in Figure 3.

E. Calibrating the Subject Scale

The center of mass of every free falling object follows a parabolic trajectory, when ignoring air resistance. If we assume that an object P moves only up and down in the vertical direction and not horizontally, the height above ground $Y_{\text{P}}(t)$ of a falling object at time t is given by the ballistic equation

$$Y_{\text{P}}(t) = -\frac{g}{2} \cdot t^2 + V_0 \cdot t + Y_0. \tag{2}$$

Here, g is gravitational acceleration of earth at the point of the experiment, which depends on the geographical latitude and the height above sea level. At the location of our experiments at $52°$ N and $55\,\text{m}$ above sea level, g is approximately $9.81\,\text{m}\,\text{s}^{-2}$ according to the International Gravity Formula 1980 [14].

Now assume that we observe the falling object using a vertically aligned, rectilinear camera, as described in section IV. Then, the object's trajectory $\mathbf{y}(t)$ in image coordinates will also be a parabola. By tracking the object with a suitable computer vision algorithm, we obtain a measured image coordinate $y_{\text{m}}(T_i)$ at each video frame timestamp T_i.

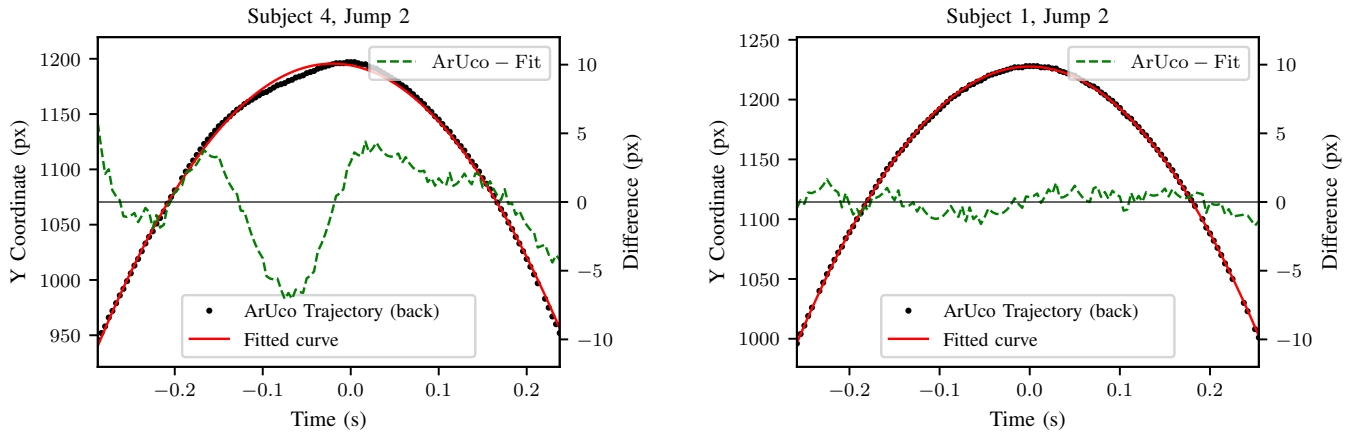

Fig. 3. The free-fall phases of two jumps from two different subjects, showing artifacts in the trajectory. The tracked trajectory • was obtained from the ArUco marker placed on the lower back of the subject. Each data point represents one frame of the high speed video. The fitted curve — was obtained by least squares regression. The dashed line - - shows the difference between the fitted curve and the tracked trajectory. (a) The left graph (subject 4) shows artifacts around the apogee of the parabola, because the belt on which the marker was attached moved together with the soft body tissue above the hip, due to the acceleration of the jump. The root mean square error (RMSE) is 3.1 px. (b) The right graph (subject 1) shows nearly no artifacts because the subject has less body fat, which causes the belt with the marker to move less during the jump. The RMSE of this fit is 0.7 px.

If we now fit a parabola to the measured pixel coordinates $y_{\text{m}}(T_i)$, we obtain a generic parabola description

$$y_{\text{fit}}(t) = a_{\text{fit}} t^2 + v_{\text{fit},0} t + y_{\text{fit},0}. \tag{3}$$

Because a has the units $[a] = \text{px}\,\text{s}^{-2}$ and the perspective transformation was linear, we know that there must be a scale factor k such that

$$a \cdot k = -\frac{g}{2} \quad \text{with units} \quad [k] = \frac{\text{m}}{\text{px}}. \tag{4}$$

So for any lengths L measured at the same distance to the camera as our initial object P, we can convert directly between pixels and meters using he formula

$$L_{\text{px}} \cdot k = L_{\text{m}}. \tag{5}$$

The advantage of this method is that it is unnecessary to calibrate the internal or external camera parameters beforehand: The parabolic trajectory of the jump itself is used to calibrate the scale for measuring the height of the jump. However, as detailed in section IV, the following conditions have to be met for the calibration to produce reliable results:

- the image sensor is parallel to gravity,
- the object-to-camera distance does not change, and
- the camera produces a rectilinear image (no lens distortion).

As noted before, this calibration procedure has only one degree of freedom: The distance Z_c between the subject plane and the focal point of the camera, which ultimately determines k, together with the intrinsic camera parameters. This makes the procedure more robust, which allows us to determine k from a single parabola, and thus calibrate each jump individually.

F. Calculation of Jump Height

Using the length conversion equation (5) it is now possible to calculate the actual real-world jump height H_{jump} in meters, as illustrated in Figure 1. Using the standing height h_0 and the apogee of the jump h_{max} in image coordinates, we can calculate H_{jump} as:

$$H_{\text{jump}} = k \cdot (h_{\text{max}} - h_0). \tag{6}$$

The value of h_0 is already known from step V-C, so we only need to calculate h_{max} from the fitted parabola coefficients a, b, c as follows:

$$h_{\text{max}} = -\frac{b^2}{4a} + c \tag{7}$$

This calculation is consistent with the definition of jump height described by Bobbert et al. [1], who also define the achieved jump height as the height difference of the body center of mass between the standing pose and the highest point of the jump.

G. Optional: Detection of invalid jumps

In order to detect invalid jump attempts, one can check how well the tracked trajectory corresponds to the fitted parabola. Any deviation is an indication that something went wrong during the jump attempt. Possible errors include:

- Tracking algorithm artifacts (incorrect position estimation, jitter).
- The marker moved relative to the subject's center of mass (e.g. loose tape, or attached to clothing).
- The subject's center of mass moved relative to the marker (e.g. moving body fat, or raising the arms or legs).
- The subject performed illegal movements (e.g. moving the head when using face tracking).

Figure 3 shows two example trajectories overlaid with the fitted parabola for two different subjects. In Figure 3 (a) one

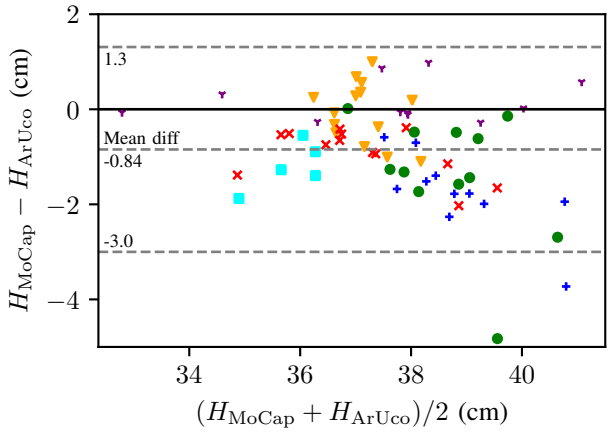

Fig. 4. Bland-Altman plot comparing the jump height calculated using the proposed algorithm on the ArUco marker near the sacrum ('ArUco') against the jump height determined with the motion capture system ('MoCap'). The dotted lines represent the 95 % limits of agreement. Test subjects are distinguished using color and symbol, showing within-subject clustering.

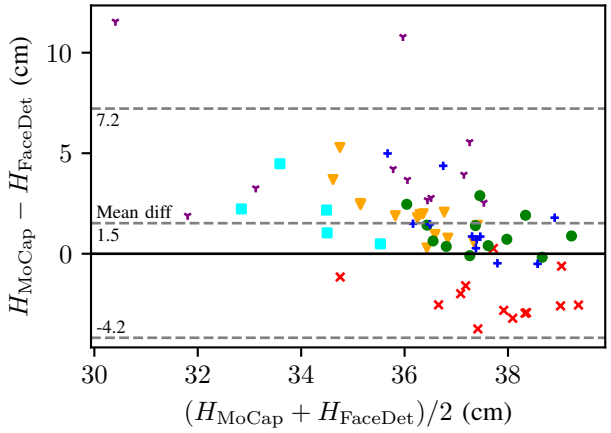

Fig. 5. Bland-Altman plot comparing the jump height calculated using the proposed algorithm on the face detection trajectory ('Face') against the jump height determined with the motion capture system ('MoCap'). The dotted lines represent the 95 % limits of agreement. Test subjects are distinguished using color and symbol, showing within-subject clustering.

can clearly see the artifacts caused by improper attachment of the marker to the subject. Manual inspection of the video footage revealed that the marker belt was attached to a soft portion of the body, which moved up and down due to the launch acceleration.

Rejecting such jumps could conceivably improve the overall accuracy of the jump height measurements.

VI. EVALUATION

A small proof-of-concept study with 6 subjects was performed to assess the feasibility of the proposed method, as described in Section III.

For every jump we took three simultaneous jump height measurements: H_{ArUco} from the ArUco marker near the sacrum, H_{FaceDet} from the face detection trajectory, and the reference height from the motion capture system H_{MoCap}. The values of H_{MoCap} were obtained by subtracting the initial standing height from the peak height of the reflective motion-capture marker near the sacrum, as described by [12]. The algorithms from Sections V-B and V-C were used to obtain these two heights from the marker trajectory.

One camera failed during the test of the last subject, so we only have 5 complete measurements of that subject. In total, we compared 66 jumps, each measured with all three methods.

Since the study was performed with a small sample size, the presented results should be interpreted with care.

VII. RESULTS

The three jump height measurements in an incomplete hierarchical design (random selected subjects with serial repeated measures) were compared by assessing Agreement in method comparison studies with replicate measurements using a mixed effect model by Carstensen et al. 2008 using of the CRAN package MethComp [15].

For H_{MoCap} vs. H_{ArUco}, the proposed method over-estimates jump height by $0.85\,\mathrm{cm}$ with a standard deviation of $1.08\,\mathrm{cm}$. The 95 % limits of agreement are $-3.0\,\mathrm{cm}$ and $+1.3\,\mathrm{cm}$, as shown in Figure 4. The intraclass correlation coefficient with a two-way mixed-effects model shows good consistency between with H_{MoCap} and H_{ArUco} with $\mathrm{ICC}(3,1) = 0.8$.

As expected, the face detection approach performed worse when compared with H_{MoCap}, with a bias of $1.52\,\mathrm{cm}$ and a standard deviation of $2.85\,\mathrm{cm}$. The 95 % limits of agreement are $-4.2\,\mathrm{cm}$ and $+7.2\,\mathrm{cm}$, as shown in Figure 5. Accordingly, the $\mathrm{ICC}(3,1) = 0.21$ only shows a weak consistency for H_{FaceDet}.

Video inspection of the outliers from Figure 5 showed that the subjects did not adhere to the test protocol in these cases and moved their head up or down during the jump. However, even when neglecting the outliers, limits of agreement are still larger than those of H_{ArUco}.

For automated detection and rejection of outliers, one might examine the root mean square error (RMSE) between the trajectory and the fitted parabola, as shown in Figure 3. However, our data shows only a weak correlation $r = 0.2$ between the RMSE and the absolute jump height difference $|H_{\mathrm{MoCap}} - H_{\mathrm{ArUco}}|$.

VIII. CONCLUSIONS AND OUTLOOK

The proposed algorithm provides an easy-to-use and inexpensive way to measure vertical jump height. The only required equipment is a smartphone capable of recording video with a short exposure time, and a method to align the phone camera vertically, such as a tripod. Slow-motion video at more than 60 fps can slightly improve performance, but is not strictly necessary and considerably lengthens the image processing time. A first evaluation with 66 individual jumps showed a good accuracy of a few centimeters. Because of the small sample size, further studies are necessary to reliably assess the accuracy of the system.

A trade-off between ease-of-use and accuracy can be made by deciding whether to track the subject's face, or a specially applied marker on the lower back. Any marker which is reliably trackable with computer vision algorithms can be used, as long as it is attached to the subject in a way that prevents relative movement. Further methods to obtain a tracking trajectory from the video could be examined in the future, for example by determining the body silhouette or estimating the positions of all body segments.

Exploratory analysis of the data revealed some degree of within-subject clustering of the measurement error, as apparent in Figures 4 and 5. Further examination of the data and the video sequences might reveal the source of this individual bias, and might reveal ways to detect or correct the underlying errors, and possibly correct the measurement protocol and provide corrective feedback to the test subjects.

Another source of errors could be the offline analysis, where subjects performed a series of jumps without direct feedback, which are analyzed later. If real-time performance and a better error classifier than RMSE (see above) can be achieved, it might be possible to identify mistakes live during a test run. Allowing the subject to retry the offending measurements could improve the overall accuracy of the system.

Future work should focus on improving the method and eliminating possible sources of measurement errors. In particular, some of the assumptions made in this study should be tested: It is not clear how precisely the axes of the accelerometer are aligned with the axes of the camera coordinate system. Depending on those findings, it might be possible to apply a keystone correction to the camera image based on accelerometer data. This would further simplify the setup, because precise vertical alignment of the smartphone would no longer be necessary. Also, a quantitative analysis of the smartphone manufacturer's camera calibration could be beneficial, in order to reduce possible distortion errors.

ACKNOWLEDGMENT

The authors would like to thank Prof. L. A. Hothorn for his assistance with the statistical modeling and analysis.

REFERENCES

[1] M. F. Bobbert and G. J. van Ingen Schenau, "Coordination in vertical jumping," *Journal of biomechanics*, vol. 21, no. 3, pp. 249–262, 1988.

[2] L. F. Aragón, "Evaluation of four vertical jump tests: Methodology, reliability, validity, and accuracy," *Measurement in physical education and exercise science*, vol. 4, no. 4, pp. 215–228, 2000.

[3] C. Balsalobre-Fernández, C. M. Tejero-González, J. del Campo-Vecino, and N. Bavaresco, "The concurrent validity and reliability of a low-cost, high-speed camera-based method for measuring the flight time of vertical jumps," *The Journal of Strength & Conditioning Research*, vol. 28, no. 2, pp. 528–533, 2014.

[4] P. F. Sturm and L. Quan, "Camera calibration and relative pose estimation from gravity," in *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, vol. 1. IEEE, 2000, pp. 72–75.

[5] Z. Zhang, "Camera calibration with one-dimensional objects." *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 7, pp. 892–9, Jul 2004.

[6] F. Wu, Z. Hu, and H. Zhu, "Camera calibration with moving one-dimensional objects," *Pattern Recognition*, vol. 38, no. 5, pp. 755–765, 2005.

[7] F. Qi, Q. Li, Y. Luo, and D. Hu, "Constraints on general motions for camera calibration with one-dimensional objects," *Pattern Recognition*, vol. 40, no. 6, pp. 1785–1792, 2007.

[8] ——, "Camera calibration with one-dimensional objects moving under gravity," *Pattern Recognition*, vol. 40, no. 1, pp. 343–345, 2007.

[9] K.-W. Chen, Y.-P. Hung, and Y.-S. Chen, "On calibrating a camera network using parabolic trajectories of a bouncing ball," in *2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*. IEEE, 2005, pp. 185–191.

[10] F. Romero-Ramirez, R. Muoz-Salinas, and R. Medina-Carnicer, "Speeded up detection of squared fiducial markers," *Image and Vision Computing*, vol. 76, 06 2018.

[11] S. Garrido-Jurado, R. Muoz-Salinas, F. Madrid-Cuevas, and R. Medina-Carnicer, "Generation of fiducial marker dictionaries using mixed integer linear programming," *Pattern Recognition*, vol. 51, 10 2015.

[12] J. S. Leard, M. A. Cirillo, E. Katsnelson, D. A. Kimiatek, T. W. Miller, K. Trebincevic, and J. C. Garbalosa, "Validity of two alternative systems for measuring vertical jump height," *The Journal of Strength & Conditioning Research*, vol. 21, no. 4, pp. 1296–1299, 2007.

[13] S. Staacks, S. Htz, H. Heinke, and C. Stampfer, "Advanced tools for smartphone-based experiments: phyphox," *Physics Education*, vol. 53, no. 4, p. 045009, May 2018. [Online]. Available: http://dx.doi.org/10.1088/1361-6552/aac05e

[14] H. Moritz, "Geodetic reference system 1980," *Journal of Geodesy*, vol. 74, no. 1, pp. 128–133, 2000.

[15] B. Carstensen, J. Simpson, and L. C. Gurrin, "Statistical models for assessing agreement in method comparison studies with replicate measurements." *The international journal of biostatistics*, vol. 4, no. 1, p. Article 16, 2008.