Markerless camera-based vertical jump height measurement using OpenPose

Fritz Webering
Leibniz University Hannover, IMS
Appelstr. 4, 30167 Hannover
webering@ims.uni-hannover.de

Holger Blume
Leibniz University Hannover, IMS
Appelstr. 4, 30167 Hannover
blume@ims.uni-hannover.de

Issam Allaham
Leibniz University Hannover
issam.allaham@stud.uni-hannover.de

Abstract

Vertical jump height is an important tool to measure athletes' lower body power in sports science and medicine. This work improves upon a previously published self-calibrating algorithm, which determines jump height using a single smartphone camera. The algorithm uses the parabolic fall trajectory obtained by tracking a single feature in a high-speed video. Instead of tracking an ArUco marker, which must be attached to the jumping subject, this work uses the OpenPose neural network for human pose estimation in order to calculate an approximation of the body center of mass. Jump heights obtained this way are compared to the reference heights from a motion capture system and to the results of the original work. The result is a trade-off between increased ease-of-use and slightly diminished accuracy of the jump height measurement.

Keywords: vertical jump height, sports, human pose estimation, convolutional neural network, gravity, parabola

1. 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 [4] is the maximum vertical displacement of the body's center of mass (CoM). However, determining the exact CoM during a continuous movement of an arbitrary person is an involved procedure, because it depends on the location and mass distribution of all body parts, including flexible tissue. The most accurate methods perform full-body motion capture of the jump and then determine the location of the CoM over time by summing the torques of all body segments as described by Aragón-Vargas [1]. However, full body motion capture is not a feasible option for everyday sports applications, due to the high cost, space, and time requirements.

Because of this, most methods do not measure jump height directly, but instead measure a different quantity and compute the jump height from that. Several such methods exist, but each has its own limitations:

- Integrating launch force over time using a force plate. – *Limited to places where an expensive, heavy force place can be carried.*
- Measuring flight time using any of various floor contact detection methods. – *Flight time may be distorted by outstretched feet during launch and landing*. [10]
- Measuring flight time through the analysis of high speed video. – *Involves manual frame-by-frame analysis of the high speed video.* [2]
- Measuring flight time using inertial measurement units (IMUs). – *Limited measurement accuracy* [13].
- Mechanically determining the highest point reached with an outstretched arm. – *Includes the arm length in the measurement, and thus does not conform to "vertical displacement of CoM" definition*
- Analyzing the free-fall parabola of a feature in a high-speed video. – *Requires markers applied directly on skin* [17].

2. Related Work

This work extends the method proposed by Webering et al. [17], which determines jump height by analyzing the

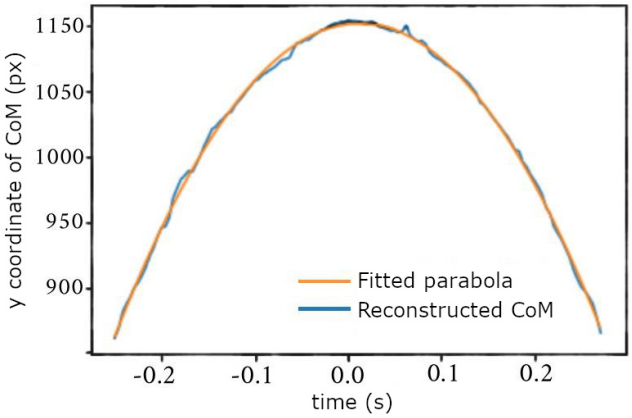

Figure 1. Example of a reconstructed CoM trace and the fitted parabola, shown in pixel coordinates. Jitter of the detected OpenPose keypoints results in the noise seen in the blue curve.

free-fall parabola of the jump in a high-speed video. Their algorithm is based on the principle that the vertical offset of a body's center of mass (CoM) describes a parabola over time while it is in free fall. When filmed with a correctly aligned camera, the y coordinate of the CoM also describes a parabola over time, when traced in pixel coordinates. The second-degree coefficient of this parabola in metric units is known to be $-\frac{g}{2}$, where g is the local gravity. The magnitude of same coefficient in pixel coordinates, obtained from a least squares fit of the parabola, can be be used to determine the scale of the image at the z depth where the parabola lies. This scale can then be used to convert the jump height in pixel coordinates into the actual jump height in metric units.

Conceptually, the algorithm is similar to the camera calibration works of Chen et al. [6], Sturm and Quan [16], Zhang [19], Wu et al. [18] and Qi et al. [12, 11]. All of these works use parabolic trajectories of falling or bouncing objects to calculate the intrinsic and extrinsic camera parameters of one or multiple cameras. However, in contrast to these works, the algorithm proposed by Webering et al. [17] simplifies the calibration problem to a single degree of freedom—the camera-to-subject distance—by placing a few constraints on the experimental setup. This allows the reconstruction of an absolute scale and calculation of the jump height from a single free-fall parabola.

Other works, such as Balsalobre-Fern et al. [2], also use high speed video to measure jump height, but with a different approach. Their method calculates the flight time by counting the number of frames between launch and landing, and not by measuring distances in the video images.

2.1. Jump Height Algorithm

The jump height calculation algorithm proposed by Webering et al. [17] consists of 6 steps, which are summarized here.

1. The vertical trajectory (y coordinate) over time of a point on the subject's body (ideally the center of mass) is obtained.
2. Starting from such a trajectory, the individual peaks (jumps) are identified in the data sequence.
3. For each peak, the standing height y_{stand} is calculated by finding a phase with little movement right before the jump.
4. The parabolic free-fall trajectory of each jump is extracted from the whole data series.
5. A parabola in image coordinates $y(t) = at^2 + v_0 t + y_0$ is fitted to the extracted data points in order to obtain the coefficient a, which determines the shape of the parabola. In real-world coordinates, this second order coefficient is known to be $a' = \frac{-g}{2}$. The two coefficients a and a' have units $\mathrm{px/s^2}$ and $\mathrm{m/s^2}$ respectively, so their ratio $k = -g/2a$ has units $\mathrm{m/px}$. Thus k can be used to convert lengths between pixels and meters in the subject plane [17]. Figure 1 shows such a parabola fitted to an example trace of the free-fall phase of a single jump.
6. The jump height in metric units is calculated using the previously determined standing height, the vertex of the parabola, and the calculated calibration coefficient as $h_{\text{jump}} = k \cdot (y_{\text{vertex}} - y_{\text{stand}})$.

3. Proposed Method

Due to the difficulty of acquiring the precise location of the center of mass (CoM), Webering et al. [17] tracked an ArUco marker [14, 8] instead, which was attached above the sacrum as shown in Figure 3. Their assumption was that this marker moves in unison with the actual CoM, because the pose of the subject while in free-fall remains mostly static: Hands on hip, legs stretched, and head held upright, gaze forward.

In this work, we propose the use of 2D human pose estimation frameworks to detect the position of the subject's limbs in the image. The OpenPose network by Cao et al. [5] is used in this work, although other 2D pose estimation frameworks could be used instead. The skeleton obtained from the pose estimation, as shown for example in Figure 2, can then be used to calculate an estimated position of the CoM in pixel coordinates by computing a weighted mean of the individual joint positions. Only the vertical coordinate of the CoM is needed for the parabola analysis, so the formula is:

$$y_{\text{CoM}} = \frac{\sum_{i=1}^{N} m_i^{\text{p}} \cdot y_{\text{seg},i}}{\sum_{i=1}^{N} m_i^{\text{p}}}, \qquad \text{with} \tag{1}$$

$$y_{\text{seg},i} = y_{\text{start},i} + l_i^{\text{p}}(y_{\text{end},i} - y_{\text{start},i}), \tag{2}$$

Segment			
Name	start — end	m^{p} (%)	l^{p} (%)
Head*	17 — 18	6.94	50.00
Trunk	1 — 8	43.46	44.86
Upper arms	2, 5 — 3, 6	2.71	57.72
Forearms	3, 6 — 4, 7	1.62	45.74
Thighs	9, 12 — 10, 13	14.16	40.95
Shanks	10, 13 — 11, 14	4.33	44.59
Hands †	—	0.61	79.00
Feet †	—	1.37	44.15

Table 1. Body segments for males, based on de Leva [7], used to calculate CoM position. The 'start' and 'end' columns refer to OpenPose keypoints, as shown in Figure 2. m^{p} is the percentage of total body mass. l^{p} is the relative position along the segment's principal axis.
* The OpenPose body_25 model has no gonion and vertex keypoints, so we use the point between the ears.
† Hand and foot segments were not used in the calculation.

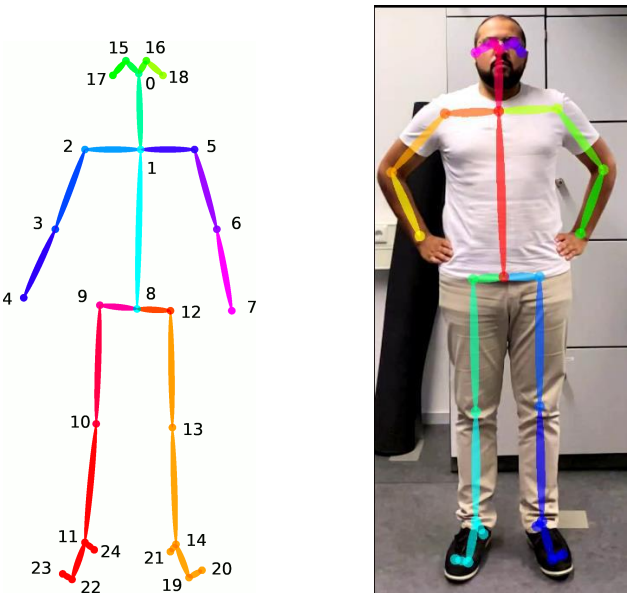

Figure 2. OpenPose body_25 skeleton, showing the keypoint numbers. Example how OpenPose maps this skeleton to a real person (segment colors carry no meaning).

where $y_{\mathrm{start},i}$ and $y_{\mathrm{end},i}$ are the y coordinates of the start and end point of each individual body segment, and m_i^{p} the proportion of the body mass represented by segment i. The parameter l_i^{p} is the relative position of the segment's CoM along the principal axis of segment i.

The coefficients m_i^{p} and l_i^{p} for the individual body segments were taken from the 1996 paper by de Leva [7] and are shown in Table 1. Only the values for males are presented in this table because all subject in the data set used in the evaluation were male. The values for females can be found in the original work by de Leva [7].

Since the OpenPose body_25 model does not contain vertex and gonion keypoints, the original values for the head segment from de Leva can not be used. We decided to use the point in the middle between the ears (keypoints 17 and 18 in Figure 2) as the CoM for the head. A manual survey of the video sequences used in the evaluation confirmed that this point is an acceptable approximation of the head's CoM calculated using de Leva's method, as long as the subject looks straight ahead, not up or down. The hand segments were omitted from the calculation because the body_25 model contains no hand keypoints, and the relative mass percentage of the hands is quite low. Furthermore, the foot segments were omitted because the detection of the foot keypoints in the available data set was unreliable.

3.1. Center of Mass Calculation

For the algorithm described in section 2.1 it is not relevant whether the actual CoM is used, or whether another point is regarded, which moves in unison with the actual CoM, with a constant vertical offset. Thus, different approaches to calculate the CoM were evaluated, in order to determine the approach which yields the most accurate jump height results. The following CoM models were considered in this work:

1. Full body CoM calculation using all segments shown in Table 1, except for hands and feet. Calculated height is denoted as $H_{\mathrm{OP,Full}}$.
2. Upper body CoM calculation using only the segments head, trunk, upper arm, and forearm. Calculated height is denoted as $H_{\mathrm{OP,Upper}}$.
3. Trunk only CoM calculation using only the trunk segment. Calculated height is denoted as $H_{\mathrm{OP,Trunk}}$.
4. Using only the neck (keypoint 1 from Figure 2) as an approximation of the CoM. Calculated height is denoted as $H_{\mathrm{OP,Neck}}$.

4. Experimental Setup

In order to evaluate the accuracy of the proposed method, we used the data set of the trial performed by Webering et al. [17]. The study included 6 healthy male participants, aged 25-35.

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.

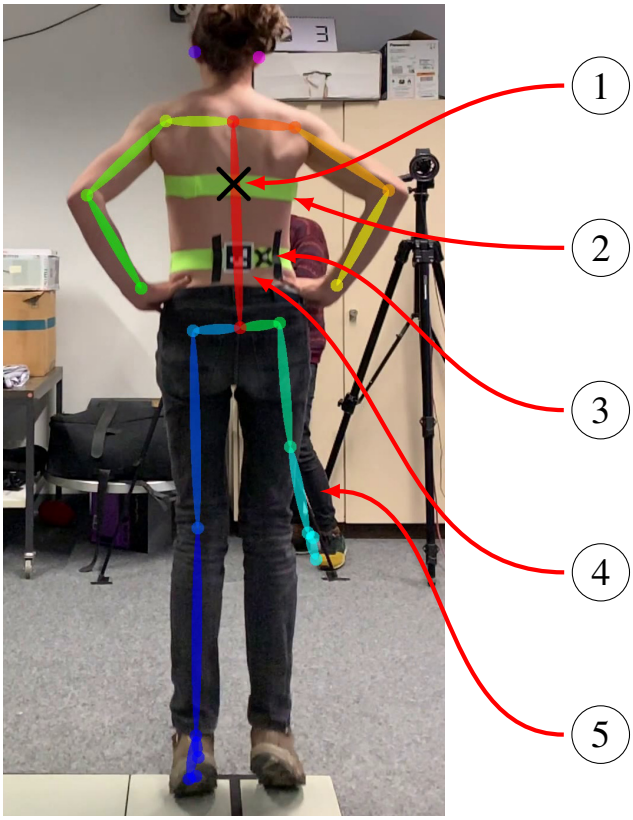

Figure 3. Example frame from one of the back videos with mislabeled leg. (1) Reconstructed CoM from upper body keypoints. (2) Strap with breast markers. (3) Back 2D ArUco marker. (4) Back 3D motion capture marker. (5) Mislabeled leg belonging to a different person.

For all trial jumps, the subject was outfitted with two retro-reflective motion capture markers and two ArUco markers [14, 8]. The ArUco markers were tracked in a high-speed video in order to obtain the free-fall parabolas necessary to determine jump height. One marker of each type was 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 3.

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 Aragón-Vargas [1], 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. A single motion capture marker, attached above the sacrum, near the center of mass, was used to calculate the reference heights. Webering et al. 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. They verified this by analyzing the parabola coefficients of the parabolic motion capture trajectory. Leard et al. [9] 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, thus minimizing motion blur.

The smartphones were aligned vertically, with the image plane parallel to the direction of gravity, using the inclination tool in the smartphone app phyphox [15]. This approach assumes that the accelerometer is calibrated in such a way, that its vertical axis is parallel to the image sensor.

4.1. Camera Calibration

Webering et al. verified that the video images obtained from the smartphone cameras were already rectified. Further checkerboard calibration yielded no improvement, so the images were processed without further correction steps applied. A standard pinhole camera model was assumed.

5. Evaluation

In order to assess the accuracy of the proposed method for calculating jump heights, all 12 videos—6 from the front camera and 6 from the back—were analyzed using OpenPose. For each video, the different CoM models were calculated as described in Section 3.1. All data series were cleaned up as described below in Section 5.1 and then further processed using the algorithm described in Section 2.1 in order to obtain the jump heights. The resulting jump heights were compared to the jump heights obtained from the motion capture system H_{MoCap} as determined by Webering et al. [17].

5.1. Data Series Cleanup

In principle, the CoM can be calculated from the OpenPose results using the weighted mean formula, described in Section 3.1. However, the analyzed data set posed some difficulties for OpenPose, such that some keypoints were mislabeled or unreliably detected in some frames. These cases would negatively affect the accuracy of the CoM calculation, so they were handles as follows:

- The video camera in front of the subject was too close to the subject, such that the legs are not visible below the knee. This made it impossible to use the whole

body model on the front videos, so only the upper body, trunk, and neck models were evaluated for the front videos.

- In the videos shot from behind the subject, another person stood close to the subject, monitoring the other video camera during the jump. This posed problems for OpenPose, which sometimes misidentified the leg of the second person as belonging to the first person. This effect caused the CoM for the whole body model to be incorrect in frames where the wrong leg was detected.

- The detection accuracy of leg and hip keypoints was greatly diminished for subjects who wore dark clothing, because of the low contrast in those areas of the image. The problem was compounded by the fact that high speed video footage is generally darker due to the lower exposure time compared to usual frame rates such as 30 or 60 fps. Since 4 of the 6 subjects in the data set were wearing black jeans, this made the lower body keypoints much more noisy than the keypoints for the rest of the body.

- Sometimes, OpenPose would randomly fail to detect arbitrary keypoints in some frames. When this happened, the respective frames were skipped and not included in the data series. When this happened for multiple frames in a row, it could negatively impact the parabola fitting.

The setup described above leads to a total of 7 jump height measurements for each of the 66 jumps in the data set: Jump heights derived from front videos are denoted as $H_{\text{OP},method,\text{F}}$ and heights derived from back videos as $H_{\text{OP},method,\text{B}}$. The measurement $H_{\text{OP,Full,F}}$ could not be computed because the legs were out of frame in the front videos. All measurements obtained this way were compared to the heights H_{MoCap}, which were determined from the motion capture traces.

For each method, a Bland Altman plot [3] is shown for visual comparison in Figures 5 and 4. The 95% limits of agreement (LOA) and the bias (mean of differences) were also calculated according to [3]. The LOA is the interval which is expected to contain $95\,\%$ of the samples and is calculated as 1.96 times the sample standard deviation of the differences between the compared methods. The $\text{ICC}_{3,1}$ were calculated for assessing agreement, classified as poor (< 0.40), fair (0.40 to < 0.60), good (0.60 to < 0.75), or excellent (≥ 0.75). ICC were calculated using R and the 'psych' library, where the two compared methods for each ICC were compared as 'raters'. Bland Altman plots, Bias and LOA were computed using Python 3.

Method A vs. Method B	LOA (cm)	Bias (cm)	$\text{ICC}_{3,1}$
H_{ArUco}† vs. H_{MoCap}	± 2.05	-0.77	0.80
$H_{\text{OP,Upper,B}}$ vs. H_{MoCap}	± 2.75	0.15	0.68
$H_{\text{OP,Trunk,B}}$ vs. H_{MoCap}	± 3.10	1.50	0.62
$H_{\text{OP,Neck,B}}$ vs. H_{MoCap}	± 3.10	-0.78	0.60
$H_{\text{OP,Upper,F}}$ vs. H_{MoCap}	± 3.40	0.79	0.57
$H_{\text{OP,Neck,F}}$ vs. H_{MoCap}	± 3.35	0.15	0.57
$H_{\text{OP,Trunk,F}}$ vs. H_{MoCap}	± 3.75	1.60	0.54
$H_{\text{OP,Full,B}}$ vs. H_{MoCap}	± 4.20	1.50	0.49

Table 2. Method comparison for the 7 OpenPose jump height measurements to the motion capture results, sorted by 95% LOA. LOA are the 95% limits of agreement as defined by Bland and Altman [3]. Bias is the mean of the differences $H_{\text{MethodB}} - H_{\text{MethodA}}$. † The H_{ArUco} values from [17] are shown for comparison only.

5.2. Results

The results for the different methods are shown in Table 2. It can be seen that the OpenPose based jump height measurements yield a slightly larger LOA than the ArUco marker based method used by Webering et al. The CoM reconstruction based on the upper body keypoints when viewed from the back yields the best LOA of all the OpenPose based measurements.

ICC values range from good for the back views (except the full body model) to fair for the front views, compared to the excellent correspondence of the ArUco based method. When compared to other jump height measurement studies like Rantalainen et al. [13], the ICC values presented here appear low compared to their ICC $= 0.97$, especially considering that Rantalainen et al. report much higher LOA than our method. This discrepancy can be explained by the lower spread in absolute jump heights of $34 - 42$ cm in the Webering et al. dataset, compared to the dataset used by Rantalainen et al., who measured jump heights between 13 and 40 cm. The smaller spread decreases the ICC as long as the absolute measurement error stays the same over whole range of the measurements.

6. Conclusion and Outlook

The presented method is a viable way to measure the vertical jump height. The usability is improved, when compared to the system proposed by Webering et al. [17], because no markers need to applied to the skin of the subject. Only a video of the person jumping is necessary to obtain a jump height measurement. Depending on the model used to reconstruct the center of mass position from the OpenPose data, the accuracy of the measured jump heights is slightly diminished, compared to the ArUco marker based method.

In order to further increase the accuracy of the presented method, different human pose estimation frameworks could

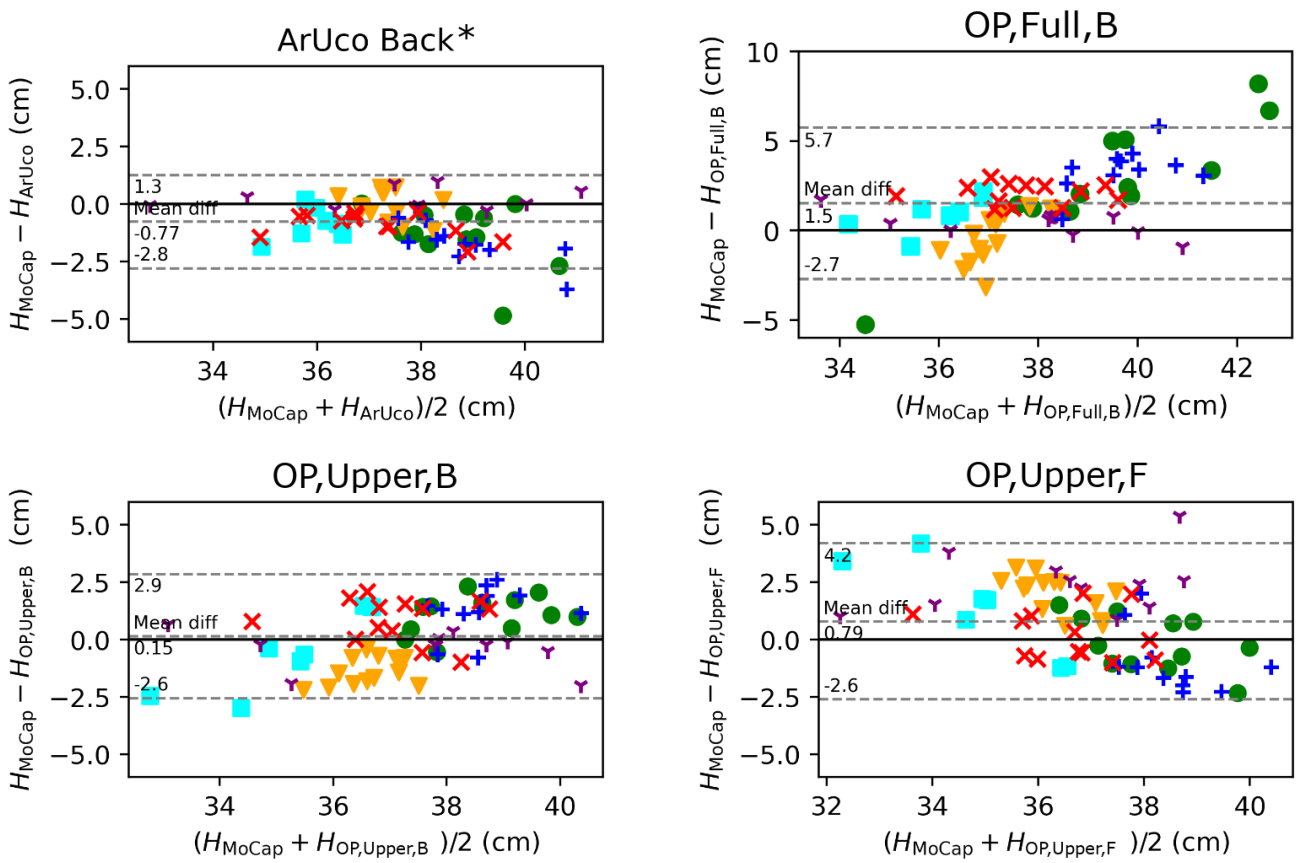

Figure 4. Bland-Altman plots for visual comparison of the different methods to the motion capture reference jump height. Dashed lines denote the mean bias, and the upper and lower 95% limits of agreement. Test subjects are distinguished using color and symbol.
* H_ArUco data from Webering et al.

be considered. As seen in Figure 1, the reconstructed CoM trajectory is still quite noisy. Using a framework which provides stabler keypoint positions might improve the parabola fitting.

References

[1] Luis F Aragón. Evaluation of four vertical jump tests: Methodology, reliability, validity, and accuracy. *Measurement in physical education and exercise science*, 4(4):215–228, 2000. 1, 4

[2] Carlos Balsalobre-Fernández, Carlos M Tejero-González, Juan del Campo-Vecino, and Nicolás 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*, 28(2):528–533, 2014. 1, 2

[3] J Martin Bland and Douglas G Altman. Agreement between methods of measurement with multiple observations per individual. *Journal of biopharmaceutical statistics*, 17(4):571–82, 2007. 5

[4] Maarten F Bobbert and Gerrit Jan van Ingen Schenau. Coordination in vertical jumping. *Journal of biomechanics*, 21(3):249–262, 1988. 1

[5] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. *IEEE transactions on pattern analysis and machine intelligence*, 43(1):172–186, 2019. 2

[6] Kuan-Wen Chen, Yi-Ping Hung, and Yong-Sheng 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*, pages 185–191. IEEE, 2005. 2

[7] Paolo De Leva. Adjustments to zatsiorsky-seluyanov's segment inertia parameters. *Journal of biomechanics*, 29(9):1223–1230, 1996. 3

[8] Sergio Garrido-Jurado, Rafael Muñoz-Salinas, Francisco Madrid-Cuevas, and Rafael Medina-Carnicer. Generation of fiducial marker dictionaries using mixed integer linear programming. *Pattern Recognition*, 51, 10 2015. 2, 4

[9] John S Leard, Melissa A Cirillo, Eugene Katsnelson, Deena A Kimiatek, Tim W Miller, Kenan Trebincevic, and Juan C Garbalosa. Validity of two alternative systems for

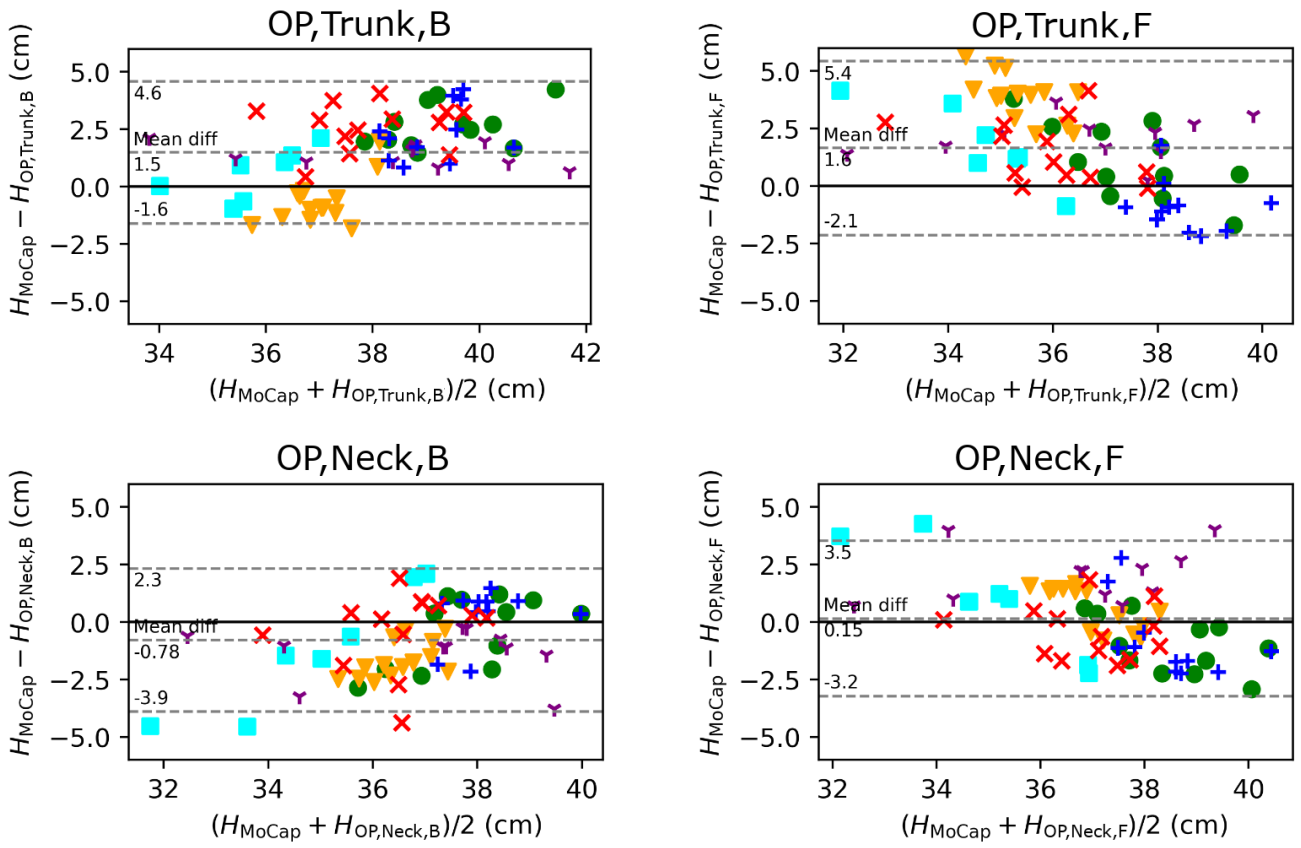

Figure 5. Bland-Altman plots for visual comparison of the different methods to the motion capture reference jump height. Dashed lines denote the mean bias, and the upper and lower 95% limits of agreement. Test subjects are distinguished using color and symbol.

measuring vertical jump height. *The Journal of Strength & Conditioning Research*, 21(4):1296–1299, 2007. 4

[10] Basilio Pueo, Patrycja Lipinska, José M Jiménez-Olmedo, Piotr Zmijewski, and Will G Hopkins. Accuracy of jump-mat systems for measuring jump height. *International Journal of Sports Physiology and Performance*, 12(7):959–963, 2017. 1

[11] Fei Qi, Qihe Li, Yupin Luo, and Dongcheng Hu. Camera calibration with one-dimensional objects moving under gravity. *Pattern Recognition*, 40(1):343–345, 2007. 2

[12] Fei Qi, Qihe Li, Yupin Luo, and Dongcheng Hu. Constraints on general motions for camera calibration with one-dimensional objects. *Pattern Recognition*, 40(6):1785–1792, 2007. 2

[13] Timo Rantalainen, Taija Finni, and Simon Walker. Jump height from inertial recordings: A tutorial for a sports scientist. *Scandinavian Journal of Medicine & Science in Sports*, 30(1):38–45, 2020. 1, 5

[14] Francisco Romero-Ramirez, Rafael Muñoz-Salinas, and Rafael Medina-Carnicer. Speeded up detection of squared fiducial markers. *Image and Vision Computing*, 76, 06 2018. 2, 4

[15] S Staacks, S Hütz, H Heinke, and C Stampfer. Advanced tools for smartphone-based experiments: phyphox. *Physics Education*, 53(4):045009, May 2018. 4

[16] Peter F Sturm and Long Quan. Camera calibration and relative pose estimation from gravity. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, volume 1, pages 72–75. IEEE, 2000. 2

[17] Fritz Webering, Leo Seeger, Niklas Rother, and Holger Blume. Measuring vertical jump height using a smartphone camera with simultaneous gravity-based calibration. In *2021 IEEE Ninth International Conference on Communications and Electronics (ICCE)*, page accepted for publication, 2021. 1, 2, 3, 4, 5

[18] FC Wu, ZY Hu, and HJ Zhu. Camera calibration with moving one-dimensional objects. *Pattern Recognition*, 38(5):755–765, 2005. 2

[19] Zhengyou Zhang. Camera calibration with one-dimensional objects. *IEEE transactions on pattern analysis and machine intelligence*, 26(7):892–9, Jul 2004. 2