# Position Estimation and Calibration of Inertial Motion Capture Systems Using Single Camera

Martin Madaras [*][||][§], Adam Riečický [||][§], Michal Mesároš [§], Michal Piovarči [‡][§], and Roman Ďurikovič [||]

[*]Institute of Software Technology and Interactive Systems, University of Technology, Vienna, Austria
[||]Department of Applied Informatics, Comenius University, Bratislava, Slovakia
[‡]Saarland University, MMCI, Saarbrücken, Germany
[§]Skeletex Research, Bratislava, Slovakia
email: madaras@skeletex.xyz
www: www.skeletex.xyz

## Abstract

The paper proposes a hybrid system for position estimation of a motion capture suit and gloves as well as a method for an automatic skeleton calibration for motion capture gloves. The skeleton calibration works with a single image scan of the hand where the skeleton is fitted. The position estimation is based on a synchronization of an inertial motion capture system and a single camera optical setup. The proposed synchronization uses an iterative optimization of an energy potential in image space, minimizing the error between the camera image and a rendered virtual representation of the scene. For each frame, an input skeleton pose from the mocap suit is used to render a silhouette of a subject. Moreover, the local neighborhood around the last known position is searched by matching the silhouette to the distance transform representation of the camera image based on Chamfer matching. Using the combination of the camera tracking and the inertial motion capture suit, it is possible to retrieve the position of the joints that are hidden from the camera view. Using the proposed hybrid technique, it is possible to capture the position even if it cannot be captured by the suit sensors. Our system can be used for both real-time tracking and off-line post-processing of already captured mocap data.

*First presented at the International Conference on Computer Graphics Theory and Applications 2018, extended and revised for JVRB*

**Keywords:** motion capture, inertial suit, optical system, optical-inertial synchronization, silhouette tracking, glove skeleton calibration

## 1 Introduction

Considering the problem of person tracking and movement analysis, there are many motion capture (mocap) solutions available in both academia and industry. Mocap is a widely used technique for storing and processing movement data of the person. Reliable motion capture and tracking are necessary for the film and games industry, virtual reality, biometrics or even healthcare.

Optical-based tracking is more problematic in the tracking area if it has occlusions. Moreover, the tracking area might be very non-convex, and therefore not coverable by optical-based mocap and tracking systems. In such a case, it is suitable to use non-optical methods for tracking; however, some other limitations appear, such as drifting, calibration and synchronization problems, or additional noise in the captured data. Sometimes, these limitations are solved by a post-processing of raw data using complex probabilistic models that have to be trained on reliable training datasets, which might be impossible to obtain in a given situation. Nowadays, there are a variety of mocap systems suitable for recording body movements.

There are two major groups of mocap systems, optical-based and inertial-based systems. Each group has its own advantages and limitations. The advantages of inertial systems are the flexible capture area (outdoor capture, water capture), occlusion independence, fast setup time, transferability and the possibility to use directly the raw output data for a 3D model. The biggest disadvantage is that one can get only rotation data for each skeleton joint. The joint positions in 3D space have to be calculated based on the calibration process and the root position estimation, which have to be approximated by a walking algorithm implemented in the mocap software.

On the other hand, the optical systems are limited to indoor use only. They have problems with occlusions and cannot directly return information about the joint rotation around the bone axis. The individual joint position can be tracked easily; however, the rotations need to be calculated in the next evaluation stage. In order to solve the positioning disadvantage of inertial systems, solutions using radio/NFER or ultrasound positioning have been proposed. However, these systems are in general hard to calibrate and synchronize.

Furthermore, in our project, we need to obtain reliable position tracking of the inertial suit using a commodity RGB camera. The whole system should be affordable and it should be compatible with both outdoor and indoor usage.

Therefore, we propose a hybrid optical-inertial system. In this system, the mocap inertial suit is combined and synchronized with a single camera. Once the camera is calibrated and the relative position and orientation are calculated, it can be used for real-time effortless position estimation. The hybrid system does not require a training phase and has advantages over both optical and inertial motion capture systems. Other hybrid tracking systems either need a complicated setup or are much more expensive. Our system requires only an inertial suit and a single RGB camera.

## 2 Related Work

**Inertial Suits** There are several inertial IMU suits available on the market: the 3DSuit by Inertial Labs, the IGS Cobra suit by Synertial, the MVN suit by XSens and the Perception Neuron suit by Noitom. The suits differ in sensor configuration, price, and precision. For example, the suits from XSens and Synertial have a higher number of sensors and can stream raw data for all the sensors. The Perception Neuron suit is a cheap and affordable solution for the general public, with a smaller set of sensors. Afterward, the streamed data available to the reader are interpolated from the raw sensor data.

**Optical Mocap** Optical systems can be divided into two main groups: systems based on passive retro-reflective or active markers, and marker-less mocap systems that are trained on a set of training images.

The marker-based systems are able to perform with much higher accuracy. In practice, optical systems and suits with markers are used, e.g. OptiTrack or Vicon. A group of multi-view RGB-only-based mocap systems working without a training stage exists, using a shape from silhouette or sums of spatial Gaussians [CBK03, SHG$^+$11]. These optical systems usually need a complicated setup and multiple cameras.

The trained probabilistic marker-less techniques [WWS09, ARS10] that work on RGB images are not very precise in general. Thus, they are typically used for academic research testing only, or they have to be fused with inertial sensors as in [PMBH$^+$10]. Recently, a helmet with two fisheye RGB cameras was proposed in [RRC$^+$16] for the motion capture of a subject wearing the helmet. The system can only capture the motion of the skeleton; this cannot be used for position tracking.

Probabilistic optical approaches can be trained directly on depth values obtained by an RGBD camera, for example Kinect. The Kinect is mostly used for real-time pose estimation [SSK$^+$13]. This probabilistic skeleton estimation is not very precise but is well suited for the fun real-time applications where Kinect tracking is mostly used. Moreover, the Kinect can be used for both real-time skeleton estimation and surface reconstruction using Kinect Fusion [IKH$^+$11]. Depth values from the RGBD camera can be used for point cloud reconstruction and the skeleton can be extracted from a point cloud. However, this process is too slow for real-time motion capture. Nevertheless, it can be used for body size estimation and calibration from a single scan. These data can be used to improve motion capture data [ASK$^+$05].

Moreover, the probabilistic optical-based systems are trained on RGB or RGBD images and estimate position in 3D space based on probabilistic models [SSK$^+$13, ARS10]. An optical flow based on Chamfer matching can be used to track the subject without a training stage [DLF06, KA08]. These methods can be used directly on the input frames; however, a background subtraction is a necessary preprocessing step to obtain robust tracking results.

An extensive comparison of inertial and optical-based motion capture suits can be found in [SN11].

**Person Tracking** Similarly as in the case of mocap solutions, the tracking can be optical-based or approximated using triangulation of distances to the signal source, e.g. GPS. The lighthouse tracking by Valve is part of HTC Vive and it is based on measuring the time delay between emitting a flash and sweeping a beam of light across the room. The receivers calculate the time delay between these two light flashes and using simple trigonometry, the 3D position of the receiver can be evaluated in real-time.

**Hybrid Systems** Several hybrid approaches were published in recent years. The hybrid systems for a skeleton and position tracking are based on a fusion of the IMU orientation data and some other sensor. In [ZKS$^+$11], the subject wearing an inertial suit is tracked by a robot with a laser scanner. Such a combination can track the subject's position and trajectory in large areas; however, it might be impossible to use the robot in small interiors, and the robot is a too expensive tool for common usage.

A fusion of multi-view RGB cameras with few IMUs was proposed in [PMBH$^+$10, vMPMR16]. These approaches for fusion give very good results; however, the fusion needs a scanned template model of the subject and the system needs multiple RGB cameras in order to correctly fit the template into the silhouette. A combination of discriminative and generative tracking using a depth sensor was used in [HMST13]. The approach also needs a template mesh model and the RGBD camera has a very limited volume in which the fusion works precisely enough.

In general, the mentioned related hybrid approaches either need a much more complicated and expensive setup (multiple cameras, depth camera, robot), or they have a much more complex tracking pipeline than our approach (template mesh scanning, non-linear energy minimization, training stage).

# 3 Optical-Inertial Synchronization

The main idea behind the optical-inertial tracking solution of the suit and the camera is determining the 3D position of the actor from his silhouette in the camera image based on his actual pose. Knowing the actor's skeleton pose from the suit in real-time, we are able to predict the body shape we are looking for within the camera image. First, a base mesh is constructed using the actor's specific parameters, such as height or local diameters. This mesh is then used for rendering a shape which is similar to the actor's silhouette in the image. A virtual camera which is used for the base mesh rendering needs to see the scene the same way as a real camera sees the scene with the actor; therefore, it needs to be calibrated.

The rendered base mesh silhouette is then used to search the local neighborhood of the last known position of the subject in the next image frame. Minimizing the energy composed of spatial integration of Chamfer matching error in the image space, we are able to perform real-time tracking of the subject. During the tracking, a 3D virtual scene is rendered and matched to the camera image; therefore, if it is matched with the precisely calibrated camera setup, we are able to directly estimate the 3D position of the subject in the real world.

## 3.1 System Overview

The whole tracking system is composed of three phases: a calibration phase, a tracking start-up phase, and an iterative tracking phase. The first two phases are used for the initial setup only to determine and to correctly represent the real world in the tracking system; therefore, the third phase is the actual tracking stage.

The calibration phase needs to be done only once, or when the camera is replaced. This step is required to acquire correct camera parameters. The parameters can be saved and reused before each tracking session.

The second stage, the tracking start-up phase, needs to be performed at least once before each session, to synchronize the real-world camera with the virtual camera of a system, and to specify the actor's starting location for the tracking. However, there is a possibility to assign these properties during the iterative tracking phase seamlessly without the interruption of the tracking procedure.

Finally, the tracking phase is iteratively performed during the whole remaining tracking time. An output of this stage is the true 3D position of the actor in both, the virtual scene and the real world.

### 3.2 Initial Setup and Calibration

For the camera calibration, an OpenCV library with its built-in modules is used. We used the ChArUco module that uses a hybrid checkerboard for both camera calibration and camera position estimation, as can be seen in Figure 1. Given several pairs of point correspondences of calibration patterns in the real world and in the image space, it is possible to find intrinsic and extrinsic parameters of the camera.

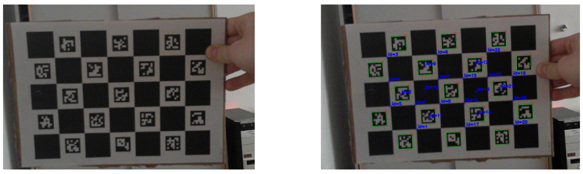

Figure 1: The ChArUco calibration board. The ChArUco is a combination of the classical calibration checkerboard and the ChArUco alphabet signs.

Next, the body size of the actor needs to be measured manually using a ruler, or in an automatic way using a Kinect or calibrated RGB camera. The measured body height and body radii are then used together with the mocap position to construct a base mesh approximating the body of the subject.

### 3.3 Virtual Scene Creation

The main idea behind the synchronization is to create a virtual scene according to the parameters acquired in the real world. This step is called the tracking start-up phase, and it must generally be executed before each tracking session when some scene properties are changed, e.g. the camera is moved or changed, or starting position is changed, etc.

In order to get the origin of the virtual space, a ChArUco board marker (see Figure 2) is placed in front of the camera. As an input, it takes corresponding points together with board parameters and as an output, it produces the rotation and translation transformations that give us a model-view matrix of the checkerboard in world space coordinates. Having the ChArUco marker detected, we are able to estimate the camera position and orientation relative to the origin. The second purpose of placing a ChArUco board into the scene is to define the starting position of the actor (the actor starts in the origin of the virtual scene). This is the position where the tracking starts. The precision of the tracked position relies deeply on the camera calibration and proper virtual scene setup (see Figure 3).

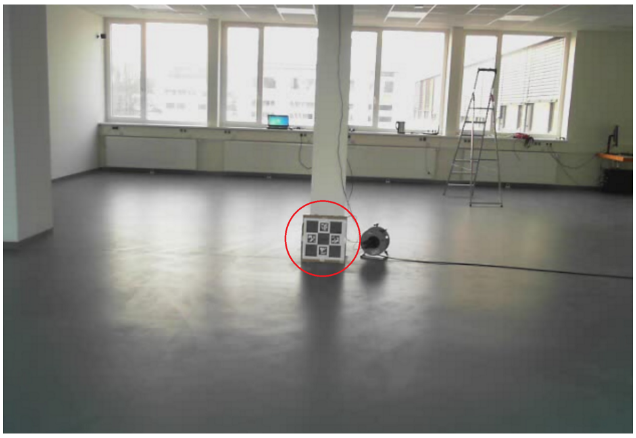

Figure 2: A model-view matrix that consists of a rotation and a translation of the ChArUco board into the camera image is obtained in the tracking start-up phase. Thus, the origin of the virtual scene is set into the ChArUco board location and the camera position and orientation are set accordingly.

If the ChArUco board cannot be used for some reason, the starting position and camera parameters can always be set manually.

### 3.4 Silhouette Image Database Construction

The camera image contains objects and subjects which are not important for the system. The goal of the system is to locate and track only the actor dressed in the motion capture suit. Therefore, the pose data from the suit are used to determine this. The reader of the suit rotations is able to stream local transformations for each frame in real-time.

Firstly, a shape that roughly represents the actor's body is needed. Here, it might be possible to use a broad set of shapes, from primitives roughly approximating the body to a highly detailed 3D scan. However, we choose to create a simple base mesh approximating the body shape mesh from the input skeleton, because it is easily customizable, scalable and can be generated in real-time for any skeleton pose. For this an SQM algorithm [BMW12] is used, which is able to generate such a mesh specifying only the skeleton and the radius of a sphere around each skeleton node (see Figure 4). These radii as well as skeleton height are dependent on the actor's body type and need to be measured or approximated manually. Such a specific base mesh is generated only once, and a pose for every frame is created by applying rotations from suit sensors and transforming the base mesh accordingly

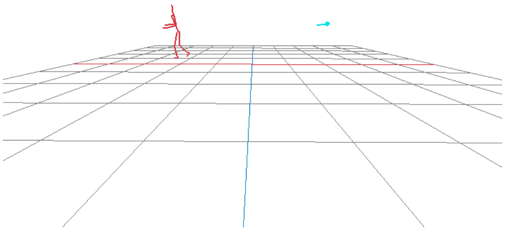

Figure 3: A virtual scene is constructed based on the camera calibration, calculated scene parameters, and current mocap data. Using either the real-time mocap data or stored off-line mocap data, a silhouette of the tracked subject is approximated and later used during the tracking phase.

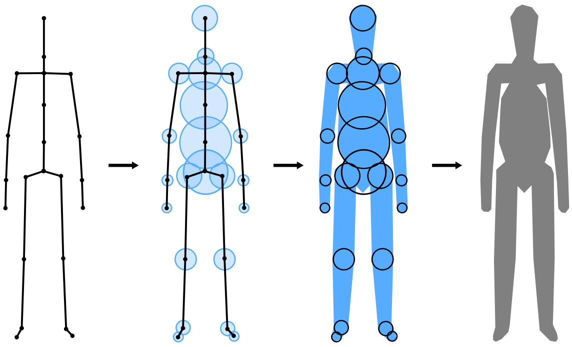

Figure 4: A base mesh is created using the skeleton acquired from the mocap suit. The skeleton is enhanced with measured radii of the actor's body. Note that the base mesh construction process is depicted in 2D only.

using a skinning algorithm.

By creating the virtual scene and calibrating it with the virtual camera, it is possible to render an image of the base mesh as if it was seen by a real camera. Afterward, this image is processed to obtain only the silhouette of a rendered image. Finally, a set of silhouette images is rendered, applying several shifts of a base mesh in eight evenly distributed directional vectors on the ground plane in 3D space. Our set consists of silhouettes rendered into images, where the skeletons of the silhouettes are shifted $k$ times by $d$ in a space around a specified pose as well as one image of a base mesh exactly in the current position, as can be seen in Figure 6. In our experiments, we used constants $d = 10cm$ and $k = 3$; therefore, in total, a set of 25 images is rendered and stored in the silhouette database for every skeleton pose.

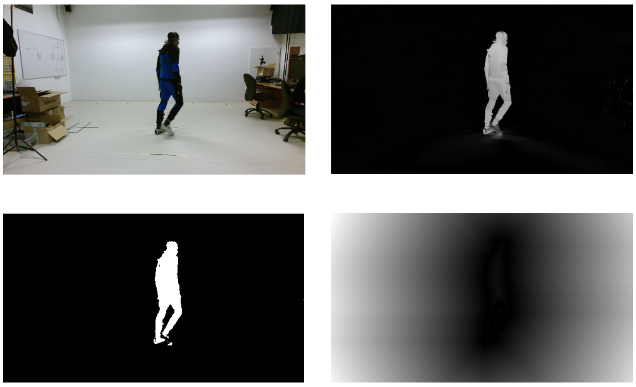

Figure 5: Tracking of the actor position using Chamfer matching. (Top) an input image and the subtracted background image in grayscale. (Bottom) applied adaptive thresholding and calculated distance transform of the silhouette.

### 3.5 Body Tracking

The tracking phase begins after the tracking start-up phase was executed successfully, which properly sets up a camera for rendering the base mesh silhouette database. An actor is located in a specified position defined by the ChArUco board placed in the scene. At this point, the tracking phase is ready to start. This start-up position is considered to be the actor's true position in the first frame. For each next frame, a motion vector is calculated to evaluate the actor's next position.

First, the captured camera image is pre-processed using background subtraction, it is thresholded and the Canny edge detection is performed so that only the actor's silhouette is obtained. Then, the image is transformed into a distance transform image (see Figure 5). Afterward, the already pre-rendered database of base mesh silhouette images is used to evaluate the energy $e$ for the optimization. The error potential $e$ is calculated for each silhouette as an integration of the distance transform function $DT(x)$ over the actor silhouette $S$ as

$$e_k = \iint\limits_S DT(\mathbf{x}(s,t))dsdt, \tag{1}$$

where parameters $s$ and $t$ are the parameters of the one-dimensional silhouette curve and a kernel function

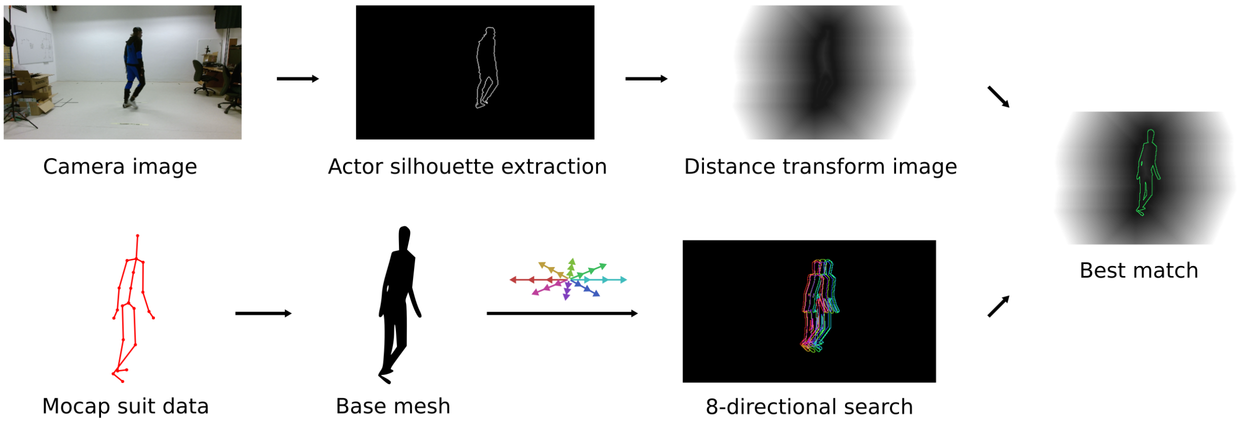

Figure 6: The pipeline of our optical-inertial synchronization. (Top) the camera image is thresholded and transformed into a distance transform image. (Bottom) a base mesh is constructed based on the acquired skeleton and it is rendered in different positions. The base of 8 search vectors is used to render 8 shifted silhouettes. The Chamfer matching of the rendered base mesh and the distance transform image is performed, and the error is evaluated. Finally, the error minimization is used to find the next position in 3D space.

that is applied over the curve to make the silhouette wider, respectively. The term $\mathbf{x}(s,t)$ is a function that maps parameters $s$ and $t$ to the image space, where function $DT(x)$ is evaluated. The integration in discrete form is performed in the image space as a sum of non-zero pixels from silhouette image $S_k$ and normalized afterward, thus the error potential integration in our implementation is calculated as

$$e_k = \frac{\sum_{i=0}^{m} \sum_{j=0}^{n} DT(i,j) \cdot S_k(i,j)}{count\_non\_zero(S_k)}, \qquad (2)$$

where $(i,j)$ refers to a pixel position of the image with dimensions $m \times n$, $DT$ is the distance transform image, $S_k$ is the binary silhouette mask image and the function $count\_non\_zero()$ returns the number of non zero pixels contained in the image. Minimizing the error energy by varying the silhouette image $S_k$, we are able to evaluate the direction and the magnitude of the subject's movement based on the shift vectors used for the construction of a database image. Adding such a vector to the position of an actor in the last frame, we are able to evaluate the actor's position in the current frame. At the end of the tracking phase, we have a raw corrected 3D position of the actor. The pipeline of the system until this point can be seen in Figure 6. To enhance the raw data, some suitable post-processing method might be used.

### 3.6 Glove Tracking

For the position estimation of the hand, we used a two-step approach instead of the one-step we used for the body position estimation. First, an initial position is estimated quickly using an image-space shape alignment. The precise position might be estimated afterward using silhouette matching similar to the one used for the body tracking. The reason for the initial alignment is, that the hand has less complex possibilities of movement, the core part of the hand does not change within the pose so rapidly, but the movement of the hand can be very fast; thus, the tracking needs to be performed with higher performance, and it has to be able to instantly recover from errors caused by the loss of position tracking. Since increasing difference between sequential camera frames decreases the performance of the Chamfer matching and the position optimization, we designed a pipeline for quick estimation of initial hand position.

The grayscale camera image of performer's hand is thresholded and post-processed using a median filtering in order to obtain a segmented real hand image. Since we used black tracking gloves we segmented the darkest area with a constant threshold value. A template mesh is deformed by a posing data obtained from the IMU sensors using linear blend skinning. At each tracked frame, the skinned template is initially rendered on a system defined default position. The position is iteratively refined by image-space alignment

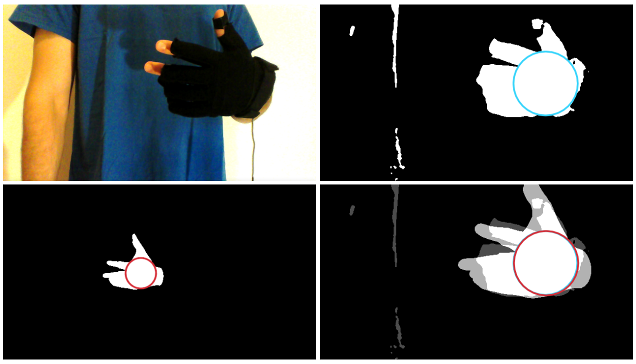

Figure 7: The pipeline of an inertial glove position estimation. (Top left) the camera image, (top right) a segmented camera image and detected blue palm circle. (Bottom left) the template model rendered at a default position with red palm circle found and (bottom right) an aligned template model with segmented camera image overlay after three iterations of image-space alignment. Use of initial default position for mesh template enables for a quick recovery from tracking failures.

procedure in order to estimate a rough position of the hand.

The image-space alignment begins with a calculation of inscribed circles for both the segmented camera image and the template image. An inscribed circle is calculated by observing global maxima of the distance transform function, where the coordinate of maxima defines the center and the value of maxima defines the radius of the circle. The two circles are then aligned in image-space for each iteration of an algorithm, using two alignment steps (the process is depicted in Figure 8). The resulting alignment should deliver a situation where the size and position of circles match.

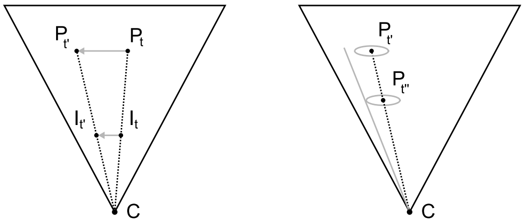

Figure 8: Two steps of image-space circle alignment procedure, a top view of a virtual camera view frustum. (Left) the positions of circle centers are aligned in order to estimate an image aligned position $P_{t'}$ of the template mesh. (Right) depth guess to position $P_{t''}$ is updated according to the ratio of circle radii.

Firstly, the circle centers are aligned in image-space: the circle centers are transformed into virtual scene space using inverse view-projection transformation with a constant depth. This gives a position of the performer's circle center $I_{t'}$ and a position of a template's circle center $I_t$ in the virtual scene. Next, an updated image aligned position of the template mesh $P_{t'}$ is calculated using formula

$$P_{t'} = P_t + \delta(I_{t'} - I_t), \tag{3}$$

where $P_t$ is the last position of the template mesh, and $\delta = d(C, I_t)/d(C, P_t)$ is the ratio of distances to the position of virtual camera $C$; $d(x, y)$ denotes the Euclidean distance between two points.

Secondly, a mesh is shifted towards/from the camera according to the ratio of template circle radius to the performer circle radius $\rho$. This allows rough estimation of the final position of performer's hand $P_{t''}$ from the image aligned position $P_{t'}$ using formula

$$P_{t''} = C + \rho(P_{t'} - C). \tag{4}$$

Note that the obtained position will not produce a perfect alignment of the two circles in image-space. This is caused by the fact, that the position of the mesh template and the position of the inscribed circle in an image-space are not in one-to-one correspondence. A procedure can be performed at multiple iterations in order to refine to the desired precision of an initial alignment.

### 3.7 Post-processing via Gaussian Filtering

The change of the estimated position of the subject in time might not be continuous. Thus, it is useful to post-process the discontinuities into a continuous movement. In our experiments, we have tried two methods of post-processing: a Gaussian-based smoothing and a Kalman filtering. We used local Gaussian smoothing in the neighborhood of 20-time steps, and the Extended Kalman Filter (EKT) [JU04] implemented in OpenCV. Using the EKT filtering, the resulting graphs seem visually smoother, but the overall error was higher. Therefore, in the final results, the Gaussian smoothing was used (see Figure 16).

## 4 Automatic Hand Calibration

In order to obtain a correct representation of a skeleton pose from inertial mocap system, a performer's body measurements are usually taken in advance. For an inertial mocap glove, we developed an automatic way of calculating bone lengths form an RGB image, using a conventional office scanner. Instead of manual measurement before mocap session, an image of the performer's hand is captured, processed by an algorithm and exported to the standard format to ensure compatibility with a variety of motion capture software.

### 4.1 Hand Image Processing

Usually, office scanners produce high-resolution images. For a better performance, we downscale the image to a predefined constant size preserving the aspect ratio of the original image. The image is converted into the HSV color space and the saturation and the value channels are used for segmentation step. The segmentation is performed in both HSV channels for higher lighting robustness. For both channels, we analyze the most frequent value according to the histogram, and if it is greater than a defined threshold, we discard the channel from the segmentation step. In case that both channels should be discarded from a process, the channel with lower value is kept. The accepted channels are segmented using threshold of a mean value and the binary masks are merged using bitwise conjunction operator. The noise in the segmentation result is reduced using morphology opening. Using this approach, we are able to perform the automatic segmentation with both darker and lighter backgrounds.

The largest segmented contour in the image is considered as a hand. However, there are cases when the subject wears a ring during scanning, which disjoints the finger from the palm. We solve the case by searching for multiple contours in the image with an area larger than an anthropometric constant and join them afterward.

In order to recognize individual fingers and their orientation in the contour, we implemented a procedure which is robust enough to handle hand images captured in different poses for both left and right hand, as depicted in Figure 9, middle:

1. the convex hull and convexity defects are calculated from the hand contour,

2. a set of four convexity defects with the largest depth are taken as the finger separation defects and the corresponding deepest points are denoted defect points,

3. the defect point with the longest corresponding line segment of the convex hull is identified as a thumb and index finger separation point $d_0$,

4. all the remaining defect points are sorted according to the distance from the thumb finger in ascending order and they are denoted as $d_1, d_2, d_3$,

5. finger contours are separated from the palm using cut lines constructed for the pairs of defect points, $(d_0, d_1)$ is used to separate the thumb, $(d_1, d_2)$ separates the index and middle finger and $(d_2, d_3)$ separates the ring and little finger,

6. finger directions are calculated by fitting a line into contours of individual fingers,

The fingertip points are calculated as an intersection of the finger directional vectors and the finger contours. The skeleton root branching template is placed according to the circle inscribed into the palm and weighted directions of the ring and middle fingers. The finger bones are estimated from fingertip points and finger directions taking advantage of human hand skeleton anthropometry (the Fibonacci sequence). The final skeleton structure can be seen in Figure 9, right.

### 4.2 Skeleton Conversion and Export

The extracted raw skeleton is defined in the image space of the scan. Next, the skeleton is scaled to the real-world size. The image space dimensions are recalculated according to the size of the A4 paper format. Finally, the skeleton file is exported according to the skeleton structure. A BVH file is used as a standard file type for the skeleton export.

## 5 Results

We demonstrate the results of our approach for capturing the motion and estimating the position of a subject in a space. The subject wears an XSens suit and the scene is captured using a Microsoft LifeCam HD 3000 RGB webcam and iDS 3 uEye monochromatic camera with a fisheye lens.

### 5.1 Correction of Mocap Suit Data

First, the position in 3D space is approximated using the standard walking algorithm usually implemented within the mocap software. In this scenario, the actor starts to run and finishes the running sequence by sliding on the ground. The sliding is the stage where the inertial mocap suit fails. We use the position estimated by XSens MVN Studio and export it into a

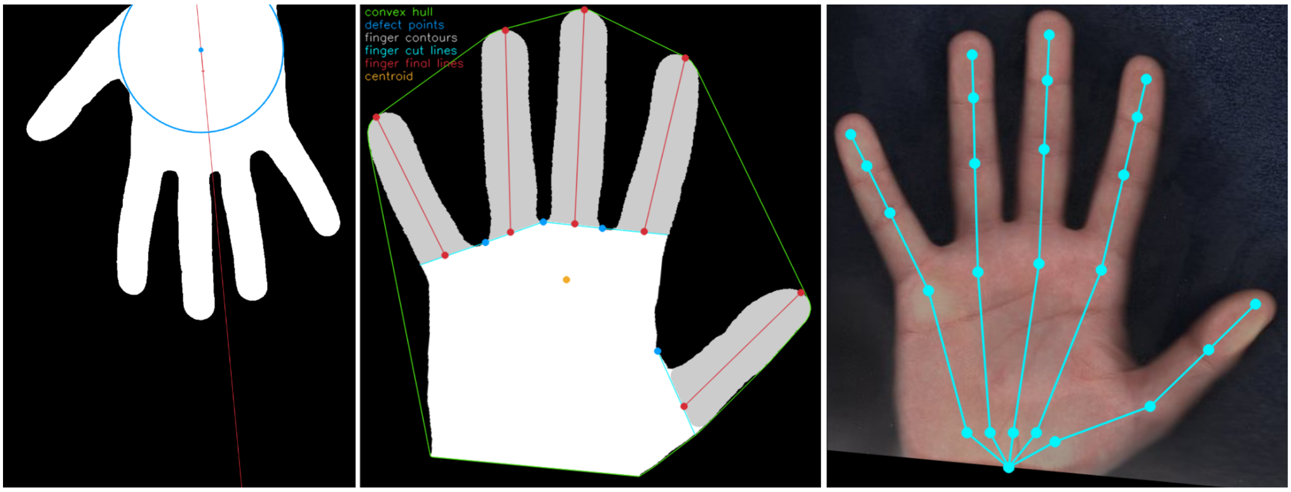

Figure 9: An automatic hand skeleton calibration. (Left) an inscribed circle and an estimated hand orientation for the skeleton root positioning, (center) a visualization of the features detected in finger recognition. (Right) the final skeleton is calculated according to the finger directions and the skeleton root.

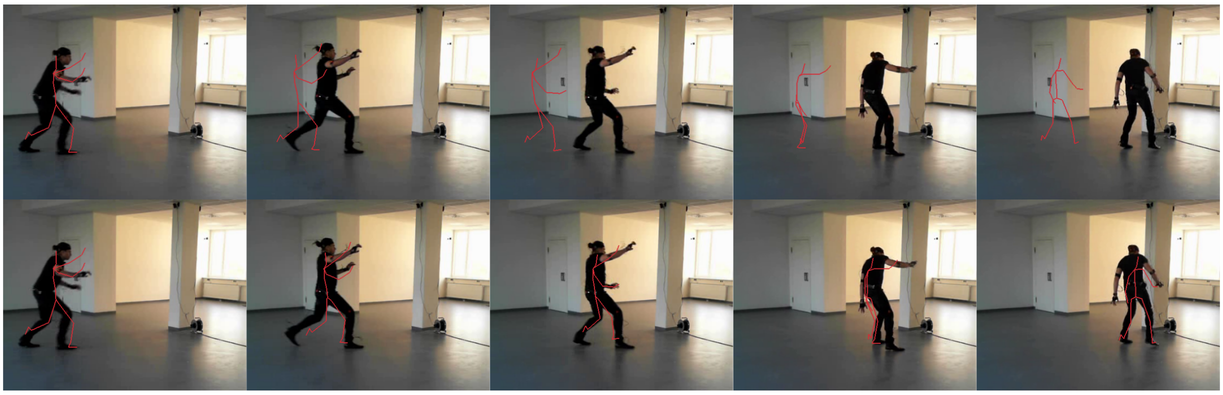

Figure 10: Comparison of (top) XSens MVN tracking and (bottom) our optical-inertial evaluation. In the second frame, the sliding starts and the mocap system fails to evaluate the position correctly. Images were captured using a Microsoft LifeCam HD 3000 RGB webcam.

BVH file. Second, the position is evaluated using our optical-inertial system. Both estimated positions, from the original method and our camera-based correction, can be seen in Figure 10.

### 5.2 Evaluation of Suit Estimated Position

In order to evaluate our method by comparing results to the ground truth, we evaluate our correction of a position inside a known environment for the movement in predefined patterns (see Figure 11). During this evaluation, the subject moves along the defined trajectories with known dimensions. During the evaluation, we track three position estimations in time. The subject is tracked by HTC Vive lighthouses (ground truth), and position estimation is done by MVN XSens Studio and our optical-inertial estimation. Comparison of the error difference in Euclidean distance of the original position from the MVN XSens Studio and our approach can be seen in Figure 14. Furthermore, we tried to smooth the raw results from our approach; the graphs with the smoothed positions of two different datasets are shown in Figure 16. In Figure 12, the setup used for the evaluation is depicted and described.

### 5.3 Extracted Hand Skeleton

The automatic skeleton extraction and calibration tool for inertial gloves works well with various scanner settings. We tested the method on a dataset of approximately 100 hand scans in different poses under various lighting conditions with positive results. A set of

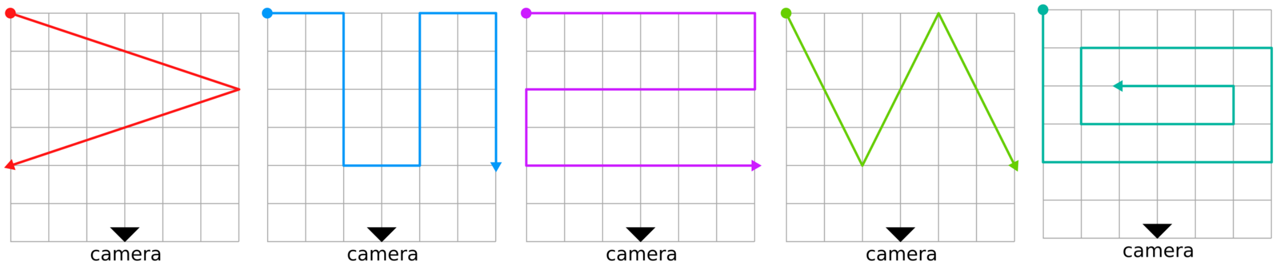

Figure 11: Evaluation of our method on different trajectories within a grid. The grid has a size of 6 × 6 meters. The patterns were chosen to fully cover the tracking area in different directions. Moreover, different types of movement were used for the evaluation, such as walking, jumping, sliding etc.

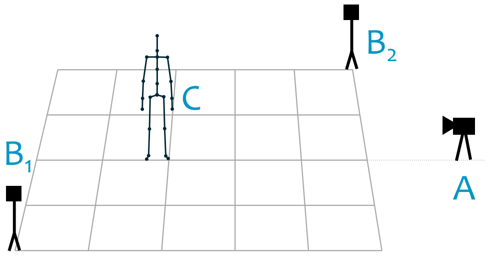

Figure 12: The evaluation setup. (C) the subject wearing a mocap suit (A) is captured by a camera (B) and tracked by HTC Vive lighthouses. The position estimated by the lighthouses is used as a ground truth when the original mocap suit tracking is compared to the proposed method.

samples from the dataset and the results can be seen in Figure 13. The extracted skeleton was imported directly into a BVH file, as well as the custom skeleton file that can be directly loaded in the Synertial glove mocap system.

### 5.4 Position Estimation of Mocap Glove

The position estimation of the glove was evaluated in a similar way as in the case of the suit. To collect the ground truth data, we used an HTC Vive tracker placed on the wrist. The overall error of hand position tracking to the ground truth was similar in comparison to the error of the full body tracking (see Figure 15). Implemented segmentation technique was sufficient for all tested cases even in situations when dark areas appeared in the camera image.

## 6 Limitations

The main limitation of the proposed solution is the dependency on the static background subtraction; thus we are not able to guarantee robust tracking in scenes with a dynamically changing background. In the case of background changes, there are edges in the image space not related to the actor that may drive the tracking into a local minimum. Another limitation of the system is the predefined set of search directions, which produces discretization errors. If required, the search space could be sampled more densely at the cost of higher computation time. The rendered base mesh is only a rough approximation of the human body; a highly detailed full-body scan could be used for better approximation of the silhouette. However, the base mesh is easy to compute, affordable to acquire and the results are good enough for our applications.

The whole tracking system of the body and the hands can perform in interactive frame-rates, 10 to 15 frames per second, using a naive implementation. Even highly real-time performance could be achieved using a more optimized code, or by performing the tracking on a set of lower resolution camera images. If the subject is fully occluded in the camera image, the image-space optimization cannot be performed and for the position calculation, only the suit data are used. Thus, in the case of full occlusion, the error of the hybrid system is the same as the inertial motion capture system only.

## 7 Conclusion and Future Work

A system for optical-inertial synchronization of the mocap suit and the camera was implemented and described in this paper. In general, the system could find its utilization in applications such as virtual reality, movement analysis, sports evaluation, and biometrics. Using a hybrid mocap system, drift issues of inertial suits can be solved. Moreover, the lack of positioning capability of inertial mocap was solved, and therefore

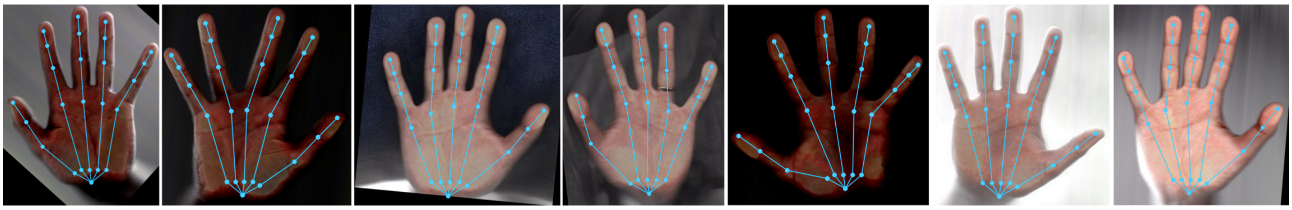

Figure 13: Extracted skeletons from hand image scans in different poses and with different lighting conditions.

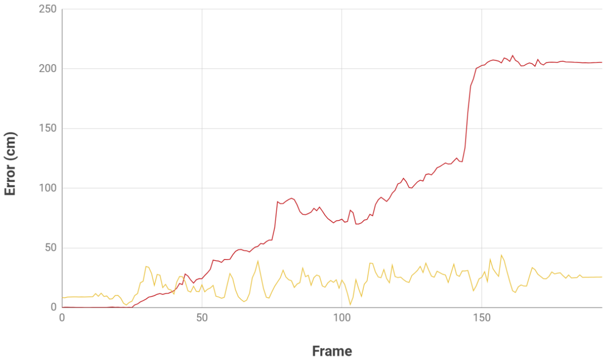

Figure 14: Comparison of error difference in Euclidean distance between the methods and ground truth obtained using HTC Vive: (red) original position from inertial suit software and (yellow) our optical-inertial method. The graph is evaluated on a dataset where the movement was the most problematic for the original method (e.g. jumping, sliding).

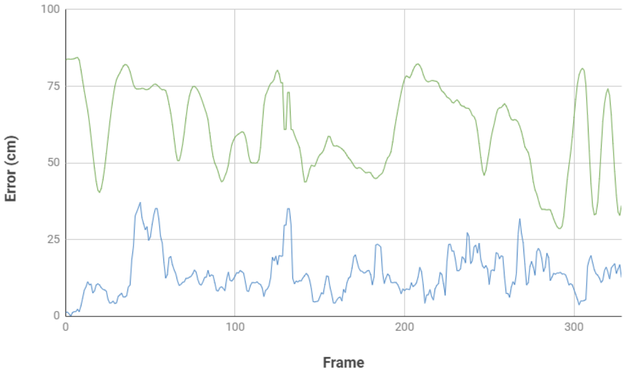

Figure 15: The evaluation of hand tracking using six iterations of image-space alignment only. (Blue) the Euclidean distance between an estimated glove position and the ground truth obtained using HTC Vive (green) a visualization of the distance to the camera which shows, that there is low correlation of the distance to the final error.

it can be directly used for subject movement analysis in 3D space, ergonomic work analysis process or virtual reality games. The inertial-optical hybrid system is capable of measuring a subject's position with high precision even if partially or fully occluded, and all the computations can be performed in real-time. These results show promising improvement for inertial suit position tracking, but more extensive evaluation is required in the future.

As future work, we would like to use the system for an automatic and effortless recalibration of the suit. The correct position and orientation of the joints, evaluated from the camera image, can be used for on-line correction of suit sensors. Moreover, extending the system for a fully automatic solution would be a good next goal for future work. The automatic body skeleton calibration could be performed using a depth scanner/camera or a more affordable, less precise solution could be achieved using RGB camera only.

## 8 Acknowledgments

We would like to thank the SAIA Action Austria-Slovakia for financial support of the project and the Interactive Media Systems group at TU Wien for all the necessary hardware and laboratory space during the project.

## References

[ARS10] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele, *Monocular 3D pose estimation and tracking by detection*, 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010, pp. 623–630, DOI 10.1109/CVPR.2010.5540156.

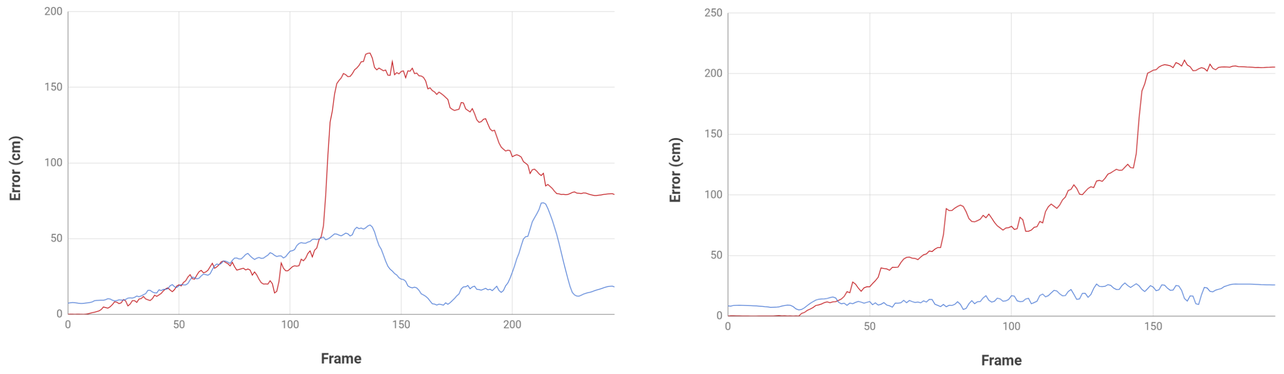

Figure 16: Comparison of error difference with filtered data: (Red) position calculated by original inertial suit software and (blue) post-processed versions of estimated positions using our approach. Another example from the datasets is plotted on the right. In both cases, the smoothed versions of evaluated positions approximate the movement and the real position of the subject much better.

[ASK+05] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis, *SCAPE: shape completion and animation of people*, ACM Transactions on Graphics **24** (2005), no. 3, 408–416, ISSN 0730-0301, DOI 10.1145/1073204.1073207.

[BMW12] Jakob Andreas Bærentzen, Marek Krzysztof Misztal, and Katarzyna Wełnicka, *Converting skeletal structures to quad dominant meshes*, Computers & Graphics **36** (2012), no. 5, 555–561, ISSN 0097-8493, DOI 10.1016/j.cag.2012.03.016.

[CBK03] German K. M. Cheung, Simon Baker, and Takeo Kanade, *Shape-from-silhouette of Articulated Objects and Its Use for Human Body Kinematics Estimation and Motion Capture*, Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR'03, IEEE, 2003, pp. 77–84, ISBN 978-0-7695-1900-5, DOI 10.1109/CVPR.2003.1211340.

[DLF06] Miodrag Dimitrijevic, Vinvent Lepetit, and Pascal Fua, *Human body pose detection using Bayesian spatio-temporal templates*, Computer Vision and Image Understanding **104** (2006), no. 2, 127–139, ISSN 1077-3142, DOI 10.1016/j.cviu.2006.07.007.

[HMST13] Thomas Helten, Meinard Müller, Hans-Peter Seidel, and Christian Theobalt, *Real-Time Body Tracking with One Depth Camera and Inertial Sensors*, 2013 IEEE International Conference on Computer Vision, ICCV, IEEE, 2013, pp. 1105–1112, DOI 10.1109/ICCV.2013.141.

[IKH+11] Sharam Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, and Andrew Fitzgibbon, *KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera*, Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, UIST '11, ACM, 2011, pp. 559–568, ISBN 978-1-4503-0716-1, DOI 10.1145/2047196.2047270.

[JU04] Simon J. Julier and Jeffrey K. Uhlmann, *Unscented filtering and nonlinear estimation*, Proceedings of the IEEE **92** (2004), no. 3, 401–422, ISSN 0018-9219, DOI 10.1109/JPROC.2003.823141.

[KA08] Itai Katz and Hamid Aghajan, *Multiple camera-based chamfer matching for pedestrian detection*, 2008 Second ACM/IEEE International Conference on Distributed Smart Cameras, 2008, pp. 1–5, DOI 10.1109/ICDSC.2008.4635734.

[PMBH+10] Gerard Pons-Moll, Andreas Baak, Thomas Helten, Meinard Müller, Hans-Peter Seidel, and Bodo Rosenhahn, *Multisensor-Fusion for 3D Full-Body Human Motion Capture*, 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 663–670, DOI 10.1109/CVPR.2010.5540153.

[RRC+16] Helge Rhodin, Christian Richardt, Dan Casas, Eldar Insafutdinov, Mohammad Shafiei, Hans-Peter Seidel, Bernt Schiele, and Christian Theobalt, *EgoCap: Egocentric Marker-less Motion Capture with Two Fisheye Cameras*, ACM Transactions on Graphics **35** (2016), no. 6, ISSN 0730-0301, DOI 10.1145/2980179.2980235.

[SHG+11] Carsten Stoll, Nils Hasler, Juergen Gall, Hans-Peter Seidel, and Christian Theobalt, *Fast articulated motion tracking using a sums of Gaussians body model*, 2011 International Conference on Computer Vision, IEEE, 2011, pp. 951–958, DOI 10.1109/ICCV.2011.6126338.

[SN11] Ståle Andreas Skogstad and Kristian Nymoen, *Comparing Inertial and Optical Mocap Technologies for Synthesis Control*, https://www.researchgate.net/publication/252066778_Comparing_Inertial_and_Optical_MoCap_Technologies_for_Synthesis_Control, 2011, Last visited January 28th, 2019.

[SSK+13] Jamie Shotton, Toby Sharp, Alex Kipman, Andrew Fitzgibbon, Mark Finocchio, Andrew Blake, Mat Cook, and Richard Moore, *Real-time Human Pose Recognition in Parts from Single Depth Images*, Communications of the ACM **56** (2013), no. 1, 116–124, ISSN 0001-0782, DOI 10.1145/2398356.2398381.

[vMPMR16] Timo von Marcard, Gerard Pons-Moll, and Bodo Rosenhahn, *Human Pose Estimation from Video and IMUs*, IEEE Transactions on Pattern Analysis and Machine Intelligence **38** (2016), no. 8, 1533–1547, ISSN 0162-8828, DOI 10.1109/TPAMI.2016.2522398.

[WWS09] Christian Wojek, Stefan Walk, and Bernt Schiele, *Multi-cue onboard pedestrian detection*, 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 794–801, DOI 10.1109/CVPR.2009.5206638.

[ZKS+11] Jakob Ziegler, Henrik Kretzschmar, Cyrill Stachniss, Giorgio Grisetti, and Wolfram Burgard, *Accurate human motion capture in large areas by combining IMU- and laser-based people tracking*, 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2011, pp. 86–91, ISBN 978-1-61284-454-1, DOI 10.1109/IROS.2011.6094430.

**Citation**

Martin Madaras, Adam Riečický, Michal Mesároš, Michal Piovarči, and Roman Ďurikovič, *Position Estimation and Calibration of Inertial Motion Capture Systems Using Single Camera*, Journal of Virtual Reality and Broadcasting, 15(2018), no. 3, February 2019, DOI 10.20385/1860-2037/15.2018.3, ISSN 1860-2037.