
This is an electronic reprint of the original article.
This reprint may differ from the original in pagination and typographic detail.

Ma, Xiaofeng; Lamine, Mohamed; Särkkä, Simo

OPTICS: Open-source Position Tracking Implementation with Consumer Smartphones

Published in:

Proceedings of the 15th International Conference on Indoor Positioning and Indoor Navigation, IPIN 2025

DOI:

[10.1109/IPIN66788.2025.11213385](https://doi.org/10.1109/IPIN66788.2025.11213385)

Published: 01/01/2025

Document Version

Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:

Ma, X., Lamine, M., & Särkkä, S. (2025). OPTICS: Open-source Position Tracking Implementation with Consumer Smartphones. In J. Nurmi, S. Lohan, A. Ometov, L. Klus, C. Mutschler, & J. Torres-Sospedra (Eds.), *Proceedings of the 15th International Conference on Indoor Positioning and Indoor Navigation, IPIN 2025* IEEE. <https://doi.org/10.1109/IPIN66788.2025.11213385>

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

OPTICS: Open-source Position Tracking Implementation with Consumer Smartphones

Xiaofeng Ma[†]
Aalto University
02150 Espoo, Finland
xiaofeng.ma@aalto.fi

Mohamed Lamine
Aalto University
02150 Espoo, Finland
mohamed.lamine@aalto.fi

Simo Särkkä
Aalto University
02150 Espoo, Finland
simo.sarkka@aalto.fi

Abstract—Motion capture systems can provide ground truth in millimeter accuracy for evaluating positioning algorithms but the high cost and specialized hardware requirements of current commercial solutions limit accessibility for many researchers. This paper presents OPTICS, an open-source, low-cost marker tracking system built using readily available hardware (smartphones and consumer-grade measuring tools). The methodological contribution of this paper is related to the implementation of the camera calibration. Particularly, we present a novel approach for estimating reference points for extrinsic calibration using pairwise distances between ground reference markers without precisely manufactured reference objects. While the system shows some accuracy limitations compared to professional setups, OPTICS offers a viable alternative for preliminary evaluations and studies without access to a high-end motion capture system.

Index Terms—marker tracking, extrinsic calibration, MDS, SMACOF, visual positioning

I. INTRODUCTION

Visual tracking systems are used in many applications, including motion capture, biomechanics research, and robotics [1]–[3]. The Vicon system [4] is a commercial system usually used as the gold standard. There are also other similar systems, such as OptiTrack [5] and Qualisys [6]. They use high-speed, high-resolution, and synchronized cameras to track the reflective markers placed on the object of interest, and calibrate the cameras by waving a well-manufactured and high-precision T-shaped wand. Although they can give sub-millimeter to 2 mm accuracy, these systems are too expensive and not affordable for many users.

Open-source motion capture systems have also been developed to provide more accessible alternatives [7]–[10]. These systems heavily rely on machine learning and computer vision techniques, thus avoiding the need for high-precision hardware. In such low-cost systems, consumer-grade cameras, such as smartphone cameras, are viable alternatives to high-end cameras for the tracking task. However, in visual motion capture systems, the extrinsic calibration of the cameras is critical, and its accuracy is fundamental for achieving accurate 3D reconstruction of the tracked object. The synchronization accuracy and speed of the low-cost cameras make their dynamic calibration (used, e.g., in Vicon) difficult.

[†]XM thanks financial support from the program of China Scholarships Council (CSC, No.202106020074) and the Research Council of Finland.

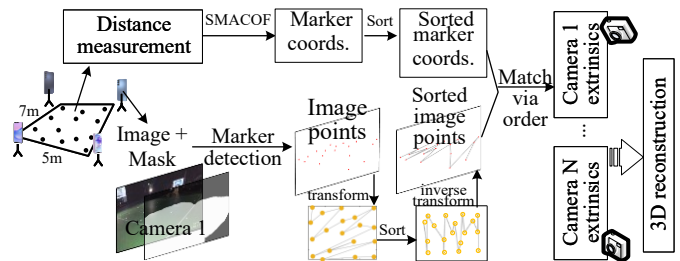


Fig. 1. The working principle of the system. Smartphones are used as synchronized cameras. The intrinsics are computed using Zhang’s method [11]. The reference points for extrinsics are estimated using the pairwise distances between the reference markers our improved SMACOF methods. The image points and the world coordinates of the reference markers are sorted and matched to each other (gray lines). The 3D reconstruction of the tracking markers is computed using triangulation.

A traditional and reliable approach for both intrinsic and extrinsic calibration is to use static reference objects to calibrate the cameras. A chessboard pattern is a common choice for camera calibration, as is done in Zhang’s method [11]. However, for extrinsic calibration, this approach is problematic because the commonly accessible size of the chessboard pattern is usually limited to A4 or A3, which is not sufficient to cover the area of interest in positioning applications. Furthermore, the method only applies to cases where only relative camera positions are needed. One can customize huge cardboards (usually several meters on the sides) as the EasyMocap [9] does, but this burdens the system economically. In this paper, we use Zhang’s method for the intrinsic calibration but propose a better alternative to the extrinsic calibration.

Our key contribution to solving this challenge is to develop improved methods for estimating the coordinates of the reference markers. The methods are based on pairwise distances between the ground markers. These distances can be measured by common tools such as measuring tapes. To convert these distances into 3D coordinates for camera calibration, we develop three methods based on the Scaling by Majorizing A Complicated Function (SMACOF) algorithm for solving the resulting Multidimensional Scaling (MDS) problem.

In this paper, we present a system called OPTICS that uses readily available consumer hardware (e.g., smartphones) to implement a visual marker tracking system. Our aim is to decrease the cost of the system as much as possible while

keeping the accuracy at a reasonable level. Furthermore, the system is fully open source. While our approach unavoidably involves a trade-off in accuracy, it provides a solution for preliminary evaluations, educational purposes, and research applications with moderate accuracy requirements.

Our contributions can be summarized as follows:

- 1) A hardware setup and open source implementation of a low-cost marker tracking system using smartphones (<https://github.com/xf-ma/OPTICS.git>).
- 2) Three improvements to the SMACOF algorithm for MDS used for extrinsic calibration, given noisy (even with missing) pairwise distance measurements, and the matching between the image points of reference markers and the world coordinates.
- 3) Evaluation of the improved methods and the whole system using simulated and real-world experiments.

The rest of the paper is organized as follows: Section II reviews the related work on motion capture systems and calibration methods. Section III describes the hardware setup and the theory behind our implementation. Section IV details our calibration methods, including our improvements on extrinsics calibration, where we propose a distance-based way to estimate the coordinates of the reference markers and improve the SMACOF algorithm for solving this problem. Section V presents simulations and experimental results. Finally, Section VI summarizes our work and discusses future directions.

II. RELATED WORK

A. Commercial Systems

Commercial motion capture systems have long been the standard for high-precision 3D tracking. Vicon, OptiTrack, and Qualisys [4]–[6] represent the current state-of-the-art marker-based optical systems. These systems typically employ multiple specialized high-speed infrared cameras placed around a capture volume, tracking reflective markers attached to subjects. They can achieve sub-millimeter accuracy under optimal conditions, with capture rates exceeding 100 Hz.

Particularly, the Vicon system [4] has become the golden standard for generating the ground truth in human motion analysis [1], robotics [2], and positioning research [3]. Its calibration process relies on proprietary wand-based techniques, where the camera parameters can be dynamically calibrated while a wand with precisely known dimensions is moving throughout the capture volume. The known dimensions will be used as constraints for computing extrinsic parameters. While these systems offer high accuracy, their cost (typically \$50,000 to \$250,000) and complex setup requirements make them inaccessible to many researchers and practitioners.

B. Open-Source Motion Capture systems

Several efforts have attempted to build more accessible open-source motion capture solutions [7]–[10]. OpenMoCap [7] is an open-source motion capture system from Brazil. Its aim is to provide an open-source implementation of a real-time motion capture system that can be used in simple animation applications without too much attention to accuracy.

However, the system uses two OptiTrack cameras, which causes the hardware cost to be high. OpenCap [8] is a recent open-source motion capture system that uses two smartphones to compute human joint angles and joint forces. It combines computer vision, machine learning, and musculoskeletal simulation and does not require other hardware (e.g., markers). The OpenCap system is designed for biomechanical analysis and does not focus on general object tracking (especially position tracking). EasyMocap [9] and FreeMoCap [10] are similar markless motion capture systems that use deep learning-based pose estimation to capture human motions.

The aforementioned systems focus mainly on human pose detection and animation applications rather than general object tracking. Their accuracy requirements are mostly at the body part level, such as hand and foot tracking. Our work differs from these systems by focusing specifically on creating an accessible alternative to commercial marker-based systems like Vicon, with particular emphasis on solving the extrinsic calibration problem using commonly available tools.

C. Multidimensional Scaling and SMACOF

Although there are open-source motion capture systems already available, they typically are based on a chessboard or reference object-based extrinsic calibration approach. However, as discussed in introduction, the approach is problematic as it requires using a huge reference object or a huge printed chessboard, which is not easy to get and may cost a lot.

Therefore, we present a novel approach to estimate the coordinates of the planar reference markers using pairwise distance measurements between them, which is an MDS problem [12]. Among MDS algorithms, SMACOF [12] has gained popularity for its monotonic convergence properties and effectiveness in handling missing data. SMACOF works by iteratively refining an initial configuration of points to minimize the stress, that is, the difference between the given distances and the distances in the estimated coordinates.

III. IMPLEMENTATION AND ALGORITHM

A. Hardware

We use the hardware that is easy to get and cheap. The necessary hardware includes:

- smartphones (≥ 2 , we use 4 Samsung Galaxy A14),
- chessboard pattern for intrinsic calibration (or any other patterns that can be easily detected by the cameras),
- reflective markers,
- distance measuring device (e.g., tape measure, or any other device that can measure distances),
- tripods or any other devices that can hold the cameras in a fixed position, and
- a computer to run the software.

The budget for the system is about \$100(+\$500 ~ \$1000), where the base cost assumes existing smartphones are used. If new phones are required, costs may increase depending on the brand and model.

B. Marker Detection

We use color detection to detect the markers in the images. Data collection is done in a low-light environment to avoid interference from ambient light. The flashlight on smartphones enhances marker reflections. Color detection is done using the RGB color space. The reflective markers will show as white and can be easily detected by RGB color detection. We remove the noisy white areas by adding a mask to the images (see illustration in Fig. 1). For example, the area highlighted by the smartphone flashlights should be blacked out, and therefore we recommend placing the smartphones outside of the moving area (e.g., on the ceiling or at a high place). Then, we detect the contours of the white areas in the images and use the center of the contours as the position of the marker. As the detection might be noisy, we remove outliers by limiting the size of the contours and the ratio of the width and height of the contours.

This is an easy and direct way to detect the marker, since we currently only support one tracking marker. We will add marker flexibility in the future.

C. Synchronization

To start video recording simultaneously on multiple smartphones, we use the software Total Control [13] to start recording through one click on the computer. The smartphones are connected to the computer via WiFi. There might be some delays in some smartphones due to the network latency but we can align the videos further through physical signals. We can use a hand clap, additional flashes of light, or sudden motions, noting that the synchronization signal should be captured by all phones. The chosen signal depends on the real task.

D. 3D Reconstruction

Once the camera parameters are known (will be explained in detail in Section IV) and the tracking marker is detected, we can compute the coordinates of the marker via triangulation.

The relationship between the homogeneous coordinates of image pixels $\mathbf{x}^p = [x, y, 1]^T$ and the homogeneous coordinates of the 3D world $\mathbf{X}^W = [X, Y, Z, 1]^T$ of the target is [14]

$$\lambda \mathbf{x}^p = \mathbf{P} \mathbf{X}^W, \quad (1)$$

where $\mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{t}]$ is the camera projection matrix, \mathbf{K} is the intrinsic matrix of the camera, and the rotation matrix \mathbf{R} and the translation vector \mathbf{t} are collectively called the extrinsic parameters of the camera. The estimation of the extrinsic camera parameters will be explained in Section IV.

Suppose we have N cameras, and the projection matrices of the cameras are $\mathbf{P}_1, \dots, \mathbf{P}_N$. For a 3D world point \mathbf{X}^W , the corresponding image points are $\mathbf{x}_1^p, \dots, \mathbf{x}_N^p$, each of which gives the relationship $\mathbf{x}_i^p \times (\mathbf{P}_i \mathbf{X}^W) = 0$, $i = 1, \dots, N$. This can be rewritten in homogeneous formulation $\mathbf{A} \mathbf{X}^W = 0$ or inhomogeneous formulation $\mathbf{A} \mathbf{X}^W = \mathbf{b}$, which can be solved by singular value composition (SVD) method $\mathbf{X}^W = \mathbf{V}_4$ and least square method $\mathbf{X}^W = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}$, respectively, where \mathbf{A} is a $2N \times 4$ matrix and computed from the projection matrices and the image points. See book [14] for more details and information.

IV. CAMERA CALIBRATION

The extrinsic parameters (extrinsics) are needed for transformation from the world coordinate system to the camera coordinate system. The intrinsic parameters (intrinsics) are needed for transformation from the camera coordinate system to the image pixel coordinate system. We proposed an approach to compute the extrinsics of the cameras using pairwise distances between reference points.

A. Intrinsic Calibration

We use Zhang's method [11] to compute the intrinsics of the cameras. This is a well-known method that uses a plane chessboard pattern with known size (i.e., known 3D coordinates of the N_c corners) for camera calibration. Since the pattern is a planar object ($Z = 0$), we can get the homography from the world plane to the image as:

$$\mathbf{H} = \mathbf{K} [\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{t}] := [\mathbf{h}_1 \quad \mathbf{h}_2 \quad \mathbf{h}_3], \quad (2)$$

where \mathbf{r}_1 and \mathbf{r}_2 are the first two columns of the rotation matrix \mathbf{R} . They are orthogonal, which gives the constraints:

$$\begin{cases} \mathbf{r}_1 \mathbf{r}_1^T = \mathbf{r}_2 \mathbf{r}_2^T, \\ \mathbf{r}_1^T \mathbf{r}_2 = 0. \end{cases} \quad (3)$$

For each image of the chessboard, the corresponding homography matrix \mathbf{H} can be computed using N_c correspondences between the image points and the 3D points.

Let $\Theta = \mathbf{K}^{-T} \mathbf{K}^{-1}$, substituting (2) into (3) and rewriting it using the columns of \mathbf{H} gives:

$$[\mathbf{v}_{12}^T \quad (\mathbf{v}_{11} - \mathbf{v}_{22})^T]^T \theta = \mathbf{0}, \quad (4)$$

where \mathbf{v}_{ij} is computed from \mathbf{H} and θ is the vectorized form of the upper triangular matrix of Θ . Stacking the equations like (4) of all images as $\mathcal{V} \theta = \mathbf{0}$, θ can be computed by the SVD method as the last column of \mathcal{V} . Then the intrinsics can be computed from θ . See the details of \mathbf{v}_{ij} , θ , and the conversion from θ to the camera parameters in article [11].

B. Extrinsic Calibration

The extrinsic calibration is the trickiest part of the system. To solve the camera extrinsics (6 DOF), at least three (the more, the more accurate) non-collinear reference points are needed. However, it is difficult to get the exact 3D coordinates of the reference points because there are no precision-manufactured objects containing orthogonal edges, and the cost of such equipment may be quite high. It is a common practice to use a paper-printed chessboard pattern as a reference object, but its easy-access size (e.g., A4) is not sufficient to cover the area of interest in a positioning application.

We use the pairwise distances between the reference points to estimate their coordinates. The reference points here are markers on the floor plane (they need to be asymmetric), that is, the height of the markers is all the same. This is a multidimensional scaling (MDS) problem [12]. If we denote the N reference points as $\mathbf{x}_1, \dots, \mathbf{x}_N \in \mathbb{R}^2$, and the pairwise distances between the reference points as a distance matrix \mathbf{D} ,

which is symmetric, the coordinates of the reference points can be estimated by minimizing the stress function:

$$\sigma(\mathbf{X}) = \sum_{i < j} w_{ij} (f(\mathbf{D}_{ij}) - \hat{d}_{ij}(\mathbf{X}))^2, \quad (5)$$

where $f(\cdot)$ is a transformation function which can be identity, linear, or monotonic, $\mathbf{X} = [\mathbf{x}_1; \dots; \mathbf{x}_N]$ are the coordinates to be estimated, \mathbf{D}_{ij} is the measured distance between points i and j , $\hat{d}_{ij}(\mathbf{X}) = \sqrt{\|\mathbf{x}_i - \mathbf{x}_j\|^2}$ is the distance between points i and j using the estimated coordinates, and w_{ij} is a weight that can be used to model the uncertainty of the distances.

1) *Traditional solution:* The classical MDS (see, e.g., [12]) gives a closed-form solution for the case when all the distance pairs are measured and the weights are 1. It is an exact solution in closed form, which is deterministic to compute and fast. However, the actual distance measurements might be noisy and/or incomplete due to data-collection challenges: the number of distances grows exponentially with the number of reference points, which is quite labour-intensive. In this case, an iterative optimization method is needed.

The SMACOF algorithm [12] is an iterative optimization method for minimizing the stress function of MDS using a majorization approach. If $f(\cdot)$ is an identity function, then (5) can be simplified to [12]:

$$\sigma(\mathbf{X}) = \sum_{i < j} w_{ij} \mathbf{D}_{ij}^2 + \text{tr}(\mathbf{X}^\top \mathbf{V} \mathbf{X}) - 2\text{tr}(\mathbf{X}^\top \mathbf{B}(\mathbf{X}) \mathbf{X}), \quad (6)$$

where $\mathbf{V} = \sum_{i < j} w_{ij} (\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^\top$, here w_{ij} can be 0 (no measurement), decimal (partial confidence based on user input or from scientific methods), or 1 (full trust or equal weighting), \mathbf{e}_i is the i -th column of the identity matrix, and

$$\mathbf{B}_{ij}^{(t)}(\mathbf{X}) = \begin{cases} \frac{w_{ij} \mathbf{D}_{ij}}{\hat{d}_{ij}(\mathbf{X})^{(t)}}, & i \neq j, d_{ij}^{(t)} \neq 0, \\ 0, & i \neq j, d_{ij}^{(t)} = 0, \\ \sum_{k \neq i} b_{ik}, & i = j. \end{cases} \quad (7)$$

We can now upper-bound (6) as [12]

$$\sigma(\mathbf{X}) \leq \text{const.} + \text{tr}(\mathbf{X}^\top \mathbf{V} \mathbf{X}) - 2\text{tr}(\mathbf{X}^\top \mathbf{B}(\mathbf{Y}) \mathbf{X}) := g(\mathbf{X}, \mathbf{Y}),$$

where \mathbf{Y} is a supporting point with known value, usually set to be the estimate from the previous iteration $\mathbf{Y} = \mathbf{X}^{(t)}$.

Now, the next iteration for estimating \mathbf{X} can be found by solving the problem: $\mathbf{X}^{(t+1)} = \arg \min_{\mathbf{X}} g(\mathbf{X}, \mathbf{X}^{(t)})$. By taking the derivative of $g(\mathbf{X}, \mathbf{X}^{(t)})$ with respect to \mathbf{X} and set it as 0, the solution can be found as:

$$\mathbf{X}^{(t+1)} = \mathbf{V}^\dagger \mathbf{B}(\mathbf{X}^{(t)}) \mathbf{X}^{(t)}, \quad (8)$$

where \mathbf{V}^\dagger denotes the pseudo-inverse of \mathbf{V} . This is known as the Guttman transform, and the aforementioned iterative method is known as the SMACOF algorithm [12].

2) *Improved solutions:* The extrinsics are critical to the accuracy of the final 3D reconstruction. Since the goal of our system is to provide ground truth, it is justified to put more effort into improving the reference point estimation.

We therefore propose three improvements to the SMACOF algorithm: 1) A gradient descent (GD) method assisted by

Algorithm 1 Improved SMACOF Algorithm

```

1: Input: Distance matrix  $\mathbf{D}$ , weight matrix  $\mathbf{W} = [w_{ij}]$ ,
   number of dimensions  $p$ , convergence threshold  $\epsilon$ , search
   interval  $[\lambda_{\min}, \lambda_{\max}]$ , maximum iterations max_iter.
2: Initialize  $\mathbf{X}^{(0)} \in \mathbb{R}^{n \times p}$  and compute matrix  $\mathbf{V}$  given  $\mathbf{W}$ 
3: for  $t = 1$  to max_iter do
4:   Compute Euclidean distances  $\hat{d}_{ij}^{(t)} = \|\mathbf{x}_i^{(t)} - \mathbf{x}_j^{(t)}\|$ 
5:   Compute matrix  $\mathbf{B}^{(t)}$  using (7)
6:   Set  $\phi \leftarrow \frac{1+\sqrt{5}}{2}$ , tol  $\leftarrow 10^{-5}$ ,  $a \leftarrow \lambda_{\min}$ ,  $b \leftarrow \lambda_{\max}$ ,
    $c \leftarrow b - \frac{b-a}{\phi}$ ,  $d \leftarrow a + \frac{b-a}{\phi}$ 
7:   while  $|b - a| > \text{tol}$  do
8:     choose method:
9:     if line search assisted gradient descent then
10:       $\mathbf{X}_c, \mathbf{X}_d \leftarrow$  use (9) with  $\alpha = c, \alpha = d$ .
11:     else if linesearch then
12:       $\mathbf{X}_c, \mathbf{X}_d \leftarrow$  use (10) with  $\alpha = c, \alpha = d$ .
13:     else if Levenberg-Marquardt then
14:       $\mathbf{X}_c, \mathbf{X}_d \leftarrow$  use (11) with  $\lambda = c, \lambda = d$ .
15:     end if
16:      $\sigma_c \leftarrow$  compute stress using (6) given  $\mathbf{V}, \mathbf{B}^{(t)}, \mathbf{X}_c$ 
17:      $\sigma_d \leftarrow$  compute stress using (6) given  $\mathbf{V}, \mathbf{B}^{(t)}, \mathbf{X}_d$ 
18:     if  $\sigma(c) < \sigma(d)$  then
19:        $b \leftarrow d, d \leftarrow c, c \leftarrow b - \frac{b-a}{\phi}$ 
20:     else
21:        $a \leftarrow c, c \leftarrow d, d \leftarrow a + \frac{b-a}{\phi}$ 
22:     end if
23:   end while
24:    $\lambda^* \leftarrow \frac{a+b}{2}$ ,  $\sigma^{(t+1)} \leftarrow$  compute stress using (6) given
    $\mathbf{V}, \mathbf{B}_{\lambda^*}, \mathbf{X}_{\lambda^*}$  according to the chosen method.
25:   if  $|\sigma^{(t+1)} - \sigma^{(t)}| < \epsilon$  or  $t \geq \text{max\_iter}$  then
26:     break
27:   end if
28: end for
29: return  $\mathbf{X}^{(t+1)}$ 

```

using a line search to find the optimal step size α ; 2) a Levenberg–Marquardt-like method to add a regularization term to the upper-bound (majorizing) function, assisted by using line search to find the optimal regularization parameter λ ; and a linesearch method where the search direction is not the GD direction, but the Guttman transform direction.

For the GD method, we search for the optimal step size in each iteration by checking if the following candidate coordinates \mathbf{X}_{try} can decrease the stress function:

$$\mathbf{X}_{\text{try}} = \mathbf{X}^{(t)} - \alpha \nabla g, \quad (9)$$

where $\nabla g = 2\mathbf{V}\mathbf{X}^{(t)} - 2\mathbf{B}(\mathbf{X}^{(t)})\mathbf{X}^{(t)}$ is the gradient of $g(\mathbf{X}, \mathbf{X}^{(t)})$ with respect to \mathbf{X} , and α is the step size. As an alternative, we propose and test a method for searching along the Guttman transform direction:

$$\mathbf{X}_{\text{try}} = \mathbf{X}^{(t)} + \alpha (\mathbf{V}^\dagger \mathbf{B}(\mathbf{X}^{(t)}) \mathbf{X}^{(t)} - \mathbf{X}^{(t)}). \quad (10)$$

For Levenberg–Marquardt method, we add a regularization

term to the majorizing function:

$$l(\mathbf{X}, \mathbf{X}^{(t)}) = g(\mathbf{X}, \mathbf{X}^{(t)}) + \lambda \text{tr}[(\mathbf{X} - \mathbf{X}^{(t)})^\top (\mathbf{X} - \mathbf{X}^{(t)})],$$

where λ is the regularization parameter. Then by minimizing the $l(\mathbf{X}, \mathbf{X}^{(t)})$ with respect to \mathbf{X} gives the following solution:

$$\mathbf{X}^{(t+1)} = (\mathbf{V} + \lambda \mathbf{I})^\dagger (\mathbf{B}(\mathbf{X}^{(t)})\mathbf{X}^{(t)} + \lambda \mathbf{X}^{(t)}). \quad (11)$$

We also search for the optimal parameters α and λ in each iteration by checking if the candidate coordinates computed by (11) decrease the stress function.

We use the golden section search method [15] to find the optimal step size λ . The improved SMACOF algorithms are described as Algorithm 1. Users may select the result based on their preferred method or the one with the minimum stress.

3) *Reference-point matching*: Since our reference markers are randomly placed, we need to find the correspondences between their image points and their corresponding 3D coordinates. The matching process is described as follows:

- compute the perspective transformation matrix \mathbf{T} between each image and the real world,
- project the image points using \mathbf{T} ,
- sort the projected points and the 3D world points via the same order (e.g., from left to right, top to bottom), and
- project the sorted image points back using \mathbf{T}^{-1} ,

where the perspective transformation matrix \mathbf{T} can be computed in a manual way or in a more automatic way.

For the manual way, we can mark at least 4 points of a rectangle on the ground, and select their image points manually in the same order. Then we can compute the transformation matrix \mathbf{T} using the 4 points by procrustes analysis [16]. The 3D coordinates of the 4 points should be roughly known but not necessarily accurate, since they are just used to recover the image from perspective projection temporarily. After the sorting, the image points will be transformed back using \mathbf{T}^{-1} . For an automatic way, we can detect the outer contour (convex hull) of the markers and use the hull points for the transformation. See an illustration in Fig. 1.

V. EXPERIMENTS

A. Evaluation of Improved SMACOF

We performed simulations in Matlab and also used real self-collected data to evaluate the performance of 4 methods: The original SMACOF, our improved SMACOF with GD, linesearch, and LM.

1) *Simulation*: We generated random markers on a 2D plane, and computed their pairwise distances. The distances were added with Gaussian noise with a standard deviation of 0.01. Some of the distances were randomly removed and assigned a weight of 0, while the rest were set to 1.

Fig. 2 shows the boxplots of each method's performance when the missing percentage is 20%, the number of markers is 12, and the random seed is from 1 to 100. Fig. 2(a) and (b) show the stress and RMSE of each method in each simulation, whose outliers are removed by the quartile methods [17]. However, the original SMACOF method still has more outliers

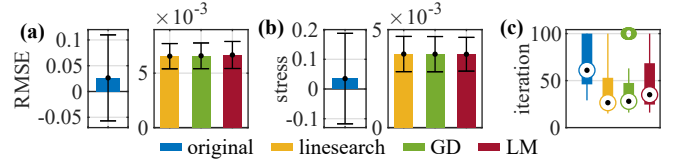


Fig. 2. Simulation results of the improved SMACOF algorithm over 100 runs (seeds 1 to 100) with 12 markers and 20% missing distances. Error bars in (a) and (b) indicate RMSE and stress, respectively, with the original SMACOF showing higher mean and standard deviation in the original SMACOF. (c) shows iterations needed to reduce stress below 0.01, with the improved method requiring roughly half as many as the original.

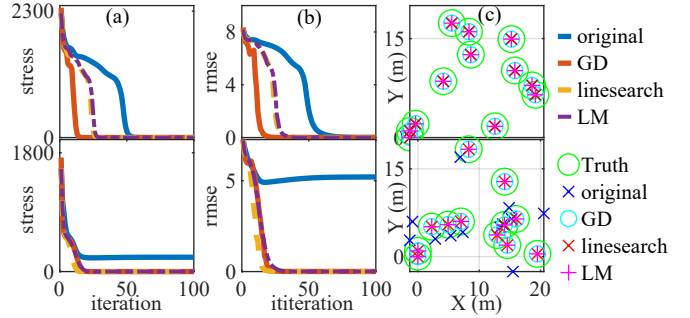


Fig. 3. The simulation of 12 markers lacking 20% distance measurements with seed 46 (first row, common case) and 47 (second row, a case where the original method failed). (a) and (b) show the stress and rmse in each iteration. (c) shows the 3D coordinates of the markers estimated by each method.

than our improved methods. Fig. 2(c) shows the number of iterations each method took to decrease the stress function to below 0.01. We can see that our improved methods take about half of the iterations of the original SMACOF method.

Fig. 3 shows the simulation when the seed is set as 46 (first row) and 47 (second row). Seed 46 is a common example case, in which all methods converge in the end and give the correct estimates of the 3D coordinates of the reference markers, although at different speeds of convergence. Seed 47 is an example case where the original SMACOF method fails to converge within 100 iterations and gives wrong estimates, while our improved methods converge in approximately 20 iterations and get a correct estimation.

2) *Real Data*: We collected real distances between 17 reference markers. All distances were recorded, but we will randomly drop some of them to simulate the missing data. To evaluate the performance of the methods with different numbers of missing data, we fix the random seed and drop different percentages of the distance measurements. The ground truth of the reference points is given by the OptiTrack system.

Fig. 4(a) shows the RMSE of each method when randomly dropping 10% to 90% of the distance data. We can see that the original SMACOF method loses significantly when the missing percentage is larger than 70%. Fig. 4(b) and (c) are the RMSE per iteration and the estimated 3D coordinates of the reference points estimated by each method when 80% distance measurements are missing. We can see that in this case, the GD and LM methods are more close to the ground truth.

It should be noted that in the simulation part V-A1, we

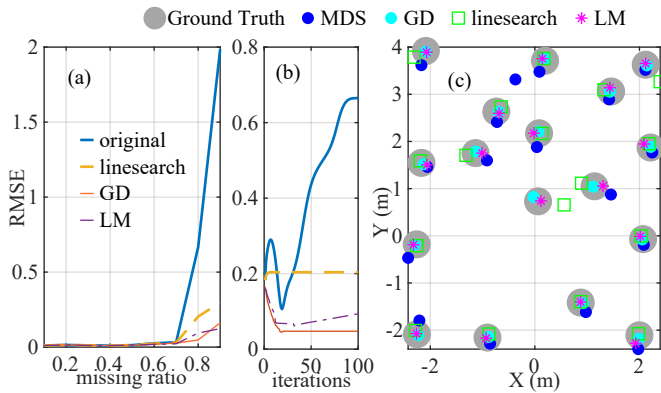


Fig. 4. Reference points estimation using real distances. (a) shows the RMSE of different missing rate (10% to 90%) of the distances. (b) shows the RMSE when missing 80% distances, and (c) is the corresponding estimation of 3D coordinates of the reference points.

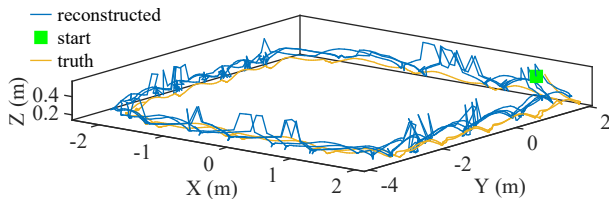


Fig. 5. The reconstructed trajectory of a marker.

initialize the coordinates randomly, while in this part, we heuristically initialize the coordinates to be closer to the real values so that all methods find the minima faster. This is completely feasible in reality, and we recommend it because a good initialization benefits the optimization.

B. 3D Reconstruction Using Real Data

Fig. 5 shows a 3D reconstruction of the trajectory of one moving marker reconstructed by our system, where the ground truth is given by the OptiTrack system. When calculating the 3D coordinates of reference points, the distance measurements missed 50% of the data, and we selected the improved method with the lowest stress. The marker was a reflective ankle bracelet worn by a person who walked along a rectangular path. Here we used the ankle bracelet to detect it even when the subject is turning. We used the DLT method to compute the 3D coordinates of the moving marker.

The figure shows that the trajectory of the moving marker shifted slightly from the ground truth. Part of the reason is that the marker used in the OptiTrack system is a small reflective ball placed near the ankle bracelet. They have a physical distance between themselves. The other reasons are that we are just using the very basic methods for the other parts of the system. In this paper, we only focus on the improvements of the methodology for estimating the coordinates of the reference points given distance measurements, as well as the complete implementation of the whole system. Improving the final accuracy will be future work.

VI. CONCLUSION

This paper presents OPTICS, an open-source marker tracking system designed to provide a viable alternative to expensive commercial solutions. The system only needs cheap and readily available devices, such as smartphones, tape rulers, and tracking markers. We addressed the critical challenge of extrinsic calibration without specialized equipment (like Vicon's wand or huge chessboard patterns). We also present novel approaches for extrinsic calibration based on complete or some pairwise distance measurements between ground markers without the need for precisely manufactured calibration objects. In particular, we propose a number of methodological improvements to the SMACOF algorithm that enhance its robustness when dealing with incomplete measurement data.

The current system is tested within a tracking area of $5 \times 7 \text{ m}^2$ with 4 smartphones. Expanding the tracking coverage to larger spaces would require additional smartphones, thereby increasing system costs. Efficient marker tracking in larger environments is a challenge, and this will be one of our future work.

REFERENCES

- [1] L. Ceriola, J. Taborri, M. Donati, S. Rossi, F. Patanè, and I. Mileti, "Comparative Analysis of Markerless Motion Capture Systems for Measuring Human Kinematics," *IEEE Sens. J.*, vol. 24, pp. 28135–28144, Sept. 2024.
- [2] S. V. Sarkisian, M. K. Ishmael, G. R. Hunt, and T. Lenzi, "Design, Development, and Validation of a Self-Aligning Mechanism for High-Torque Powered Knee Exoskeletons," *IEEE Trans. Med. Robot. Bionics*, vol. 2, pp. 248–259, May 2020.
- [3] M. Delamare, R. Boutteau, X. Savatier, and N. Iriart, "Static and Dynamic Evaluation of an UWB Localization System for Industrial Applications," *Sci*, vol. 2, p. 23, June 2020.
- [4] Vicon Motion Systems Ltd., "Vicon Motion Capture System." <https://www.vicon.com/>. Accessed: 2025-May-12.
- [5] OptiTrack Motion Systems Ltd., "Optitrack Motion Capture System." <https://www.optitrack.com/>. Accessed: 2025-May-12.
- [6] Qualisys Motion Systems Ltd., "Qualisys Motion Capture System." <https://www.qualisys.com/>. Accessed: 2025-May-12.
- [7] D. L. Flam, J. V. B. Gomide, and U. Fumec, "OpenMoCap: An Open Source Software for Optical Motion Capture," *VIII Braz. Symp. Games Digit. Entertain.*, 2009.
- [8] S. D. Uhlrich, A. Falisse, L. Kidziński, J. Muccini, M. Ko, A. S. Chaudhari, J. L. Hicks, and S. L. Delp, "OpenCap: Human movement dynamics from smartphone videos," *PLOS Computational Biology*, vol. 19, p. e1011462, Oct. 2023.
- [9] "Easymocap - make human motion capture easier." Github, 2021.
- [10] J. S. Matthis and A. Cherian, "FreeMoCap: A free, open source markerless motion capture system." Zenodo, July 2022.
- [11] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, pp. 1330–1334, Nov. 2000.
- [12] P. J. F. Groenen and M. Van De Velden, "Multidimensional Scaling by Majorization: A Review," *J. Stat. Soft.*, vol. 73, no. 8, 2016.
- [13] Total Control Ltd., "Total control." <https://www.sigma-rt.com/en/tc/>. Accessed: 2025-May-12.
- [14] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge: Cambridge University Press, second edition ed., 2003.
- [15] W. H. Press, ed., *Numerical Recipes: The Art of Scientific Computing*. Cambridge, UK ; New York: Cambridge University Press, 3rd ed ed., 2007.
- [16] I. L. Dryden and K. V. Mardia, *Statistical Shape Analysis: With Applications in R*. Wiley, 2nd ed., 2016.
- [17] A. Agresti, *Statistical Methods for the Social Sciences*. Pearson, 5th ed., 2018.