# Structural Triangulation: A Closed-Form Solution to Constrained 3D Human Pose Estimation

Zhuo Chen, Xu Zhao(✉), and Xiaoyue Wan

Department of Automation, Shanghai Jiao Tong University, Shanghai, China
{chzh9311, zhaoxu, sherrywaan}@sjtu.edu.cn

**Abstract.** We propose *Structural Triangulation*, a closed-form solution for optimal 3D human pose considering multi-view 2D pose estimations, calibrated camera parameters, and bone lengths. To start with, we focus on embedding structural constraints of human body in the process of 2D-to-3D inference using triangulation. Assume bone lengths are known in prior, then the inference process is formulated as a constrained optimization problem. By proper approximation, the closed-form solution to this problem is achieved. Further, we generalize our method with *Step Constraint Algorithm* to help converge when large error occurs in 2D estimations. In experiment, public datasets (Human3.6M and Total Capture) and synthesized data are used for evaluation. Our method achieves state-of-the-art results on Human3.6M Dataset when bone lengths are known and competitive results when they are not. The generality and efficiency of our method are also demonstrated.

**Keywords:** Multi-view 3D human pose estimation · Constrained optimization · Triangulation

## 1 Introduction

3D Human Pose Estimation (3D HPE) is a fundamental yet difficult problem in computer vision. From the perspective of sensor utilization, this problem could be divided into two categories, namely, monocular [9,17,18,26,27,33,36] and multi-view [2,6,10,16,20,25] based, both obtaining increasing attention in recent years. Different from monocular 3D HPE, multi-view systems can acquire depth information theoretically from multiple measurement instead of merely

This work has been funded in part by the NSFC grants 62176156 and the Science and Technology Commission of Shanghai Municipality under Grant 20DZ2220400. The code is available at https://github.com/chzh9311/structural-triangulation.

**Supplementary Information** The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-20065-6_40.

S. Avidan et al. (Eds.): ECCV 2022, LNCS 13665, pp. 695–711, 2022.
https://doi.org/10.1007/978-3-031-20065-6_40

training data, which is an inherent advantage. In this paper, we try to improve multi-view 3D HPE via a novel pathway.

Triangulation is a basic and common module in 3D HPE, which estimates the 3D joint positions by leveraging their 2D counterparts measured in multiple images to 3D space [13]. The module is usually used in a two-stage procedure: first estimating 2D poses in multi-view images, then applying triangulation to obtain 3D human pose [16,19,32]. Remarkable progress has been achieved under this pipeline. However, conventional triangulation methods are designed for individual points, and the associations between points are not specially considered. But human body possesses an innate structure, containing relations between joints, which can provide strong priors for 3D HPE. To overcome this shortcoming, some methods, like 3DPSM [4,23] and post-process methods [10,18] are proposed, and exceed previous methods in precision so the effectiveness of using human priors is demonstrated. But some aspects are still out of focus of these works. 3DPSM is usually time-consuming due to the large search space; post-process methods have limited effects, where priors are not naturally and thoroughly applied.

Ideally, we expect to build a grace and simple expression for the optimal 3D pose considering 2D poses, camera settings, and human priors. Thus, 2D-to-3D inference can be more efficient and accurate, while keeping the simplicity of linear triangulation. We start from a simple idea of embedding structural information in triangulation. The problem is formulated as a process to minimize weighted square re-projection error. Using predefined bone lengths, a constrained optimization problem is constructed. The solution to this problem is made closed-form by proper approximations and linearizations. It directly produces the optimal pose of a certain tree structure with predefined edge lengths. We call this novel triangulation method as *Structural Triangulation* (ST).

To make aforementioned approximations feasible, some conditions should be satisfied, but they may not always hold in practice, causing ST to diverge. So we propose the *Step Constraint Algorithm* (SCA) to promote its adaptivity. The algorithm split the optimization problem into small steps. and the optimal pose gets updated when stepping from one point to the next, meanwhile, the preconditions are satisfied at each step.

We conduct comprehensive experimental evaluation to the proposed method. First, we use the 2D backbone provided in [16] to capture 2D pose, and then the 3D pose is estimated using our method. Two public datasets, Human3.6M [15] and Total Capture [29], are used. We achieve state-of-the-art result with precise bone lengths, and promising result with bone lengths estimated from T-pose. Next, we generate multi-view 2D estimations by shifting ground truth re-projection randomly, and the test result shows that our method can work well regardless of the choice of 2D backbones. Finally, the efficiency of our method is validated by comparing the run time with previous methods.

In sum, the contributions of this work are in the following three aspects.

1. We construct a novel constrained 3D HPE problem and derive a closed-form solution called Structural Triangulation. For the first time, structural priors (bone lengths) are embedded in triangulation in a simple analytical form.
2. We design the Step Constraint Algorithm, which helps ST converge when 2D pose estimations are not precise enough.
3. We evaluate our method on Human3.6M Dataset, Total Capture Dataset, and synthesized data. The precision and efficiency of our method are validated by comparing with other state-of-the-art methods.

## 2 Related Work

Recently, many works are proposed to solve the problem pf Multi-view 3D HPE. We roughly classify them as geometry and optimization based methods.

**Geometry-Based Methods.** Epipolar geometry is the theory basis of triangulation. A two-stage framework is commonly used in multi-view 3D HPE [16,20,25]. In the first stage, 2D poses are estimated from the given views separately. Secondly, 3D pose is inferred from 2D poses by triangulation methods. This framework is straightforward, practical, and proved effective. Recent works make difference mainly in 2D estimations or 3D volume introduction. The triangulation itself, however, remain conventional as concluded in [13].

Under the framework of epipolar geometry, feature fusion is another concern in recent [6,24,25,32,35]. The method generally connects 2D backbones of different views to fuse information from all views before outputting 2D heatmaps. The effectiveness of this kind of methods in promoting both 2D and 3D estimations is validated. Further, [25] provides a light-weighted version, and [32] contributes in generalizing it.

Recently, in some works, information loss in the process of 2D pose estimation, is significantly concerned. So the 2D estimation step is eliminated and 3D pose is obtained directly from multi-view images, like fusing heatmaps directly in volumes [16] and 3D pose regression [30]. Although having the benefit of precision improvement, the computational cost is also increased.

**Optimization-Based Methods.** Early works of multi-view 3D HPE start from optimizing human pose given 2D features [12]. Although the development of deep learning offers a better solution for feature extraction, optimization is still active in recent works. Such methods generally solve the problem by designing an objective function that fuses all known information.

The most common and effective method under this pipeline, is 3D Pictorial Structure Model (3DPSM). The original PSM was first proposed in [11] to match certain structures on images. After PSM achieved promising performance in 2D pose estimation [34], 3DPSM was developed to deal with 3D HPE [2,4]. It succeeds in optimizing posterior probability given observations and human priors. But global optimization is generally implemented by grid sampling and therefore is time-consuming. To balance time and precision, recurrent PSM is proposed in [24] so a faster convergence is realized.

Besides 3DPSM, there are other ways to solve such a problem. In [10], SMPL model [3] is used to fit 3D pose by optimization so that pose and shape are reconstructed simultaneously. Shape models can eliminate some unfeasible poses. However, the increase in pose precision is quite limited because of redundant shape parameters. In [5], Maximum A Posteriori (MAP) is integrated with trust region method [8] to optimize 3D pose, but it suffers from initialization.

In conclusion, to achieve better precision, the current models become increasingly complicated with less efficiency. Moreover, current triangulation methods generally treat joints independently, yet the significance of human priors has been well proved. So in this work, we propose to produce optimal 3D pose based on a closed-form solution by utilizing human priors in a novel way.

## 3 Method

The whole framework of the proposed method is shown in Fig. 1. In this section, we focus on describing the process of formulation, while the detailed inductions can be found in Sec. 1 in Supplementary.

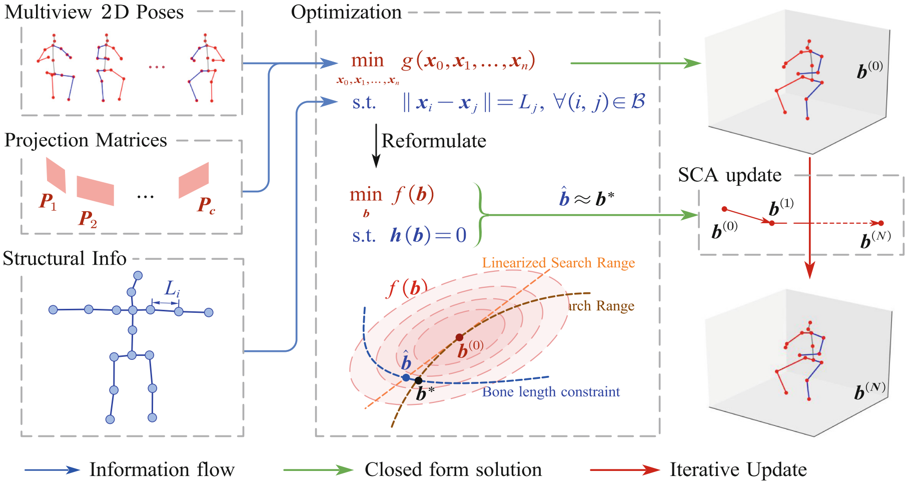


**Fig. 1.** The framework of our method. With the 2D poses and camera projection matrices, we formulate a quadratic objective function based on re-projection error, which produces the initial solution $b^{(0)}$. And bone lengths are used as constraints to relate joints. Then the unknowns are converted from joint positions to bone vectors and the problem is reformulated. In the lower middle figure, brown line is the analytical search range given by KKT condition, we linearize it so that a closed-form solution $\hat{b}$ is derived, which is close to the analytical one $b^*$. It is further used in SCA to update from $b^{(0)}$ to the final solution $b^{(N)}$. (Color figure online)

### 3.1 Problem Formulation

Given multiple images taken by several calibrated cameras, we are going to estimate the 3D human pose in scene, where only the case of a single person is considered. Suppose the 2D poses in each view, along with the lengths of body bones, could be available, obtained from other existing methods.

First of all, we model the overall human body as a tree structure with joints as nodes and bones as edges, where in total $n + 1$ joints indexed by $i = 0, 1, \dots, n$ are considered, and 0 represents the root joint (usually hip). Bones are represented in form of $(i, j)$, where $i$ is the parent of $j$. Mark the bone set as $\mathcal{B}$.

Now we can organize the known and unknown variables. The projection matrices $\boldsymbol{P}_k$ of all cameras indexed from 1 to $c$, the bone length $L_j$ of each bone $(i, j)$, the 2D location $\hat{\boldsymbol{x}}_{i,k}$ of joint $i$ on image from camera $k$, and the corresponding weight $w_{i,k}$ (usually the belief given by 2D backbones) are known. The 3D coordinates of human joints, denoted by $\boldsymbol{x}_0, \boldsymbol{x}_1, \dots, \boldsymbol{x}_n$, are unknowns and need to be determined. Then our goal is to minimize the total weighted square re-projection error with predefined values of all the bone lengths, i.e.:

$$\min_{\boldsymbol{x}_0, \boldsymbol{x}_1, \dots, \boldsymbol{x}_n} \sum_{i=0}^{n} \sum_{k=1}^{c} w_{i,k} \left\| H^{-1} \left( \boldsymbol{P}_k H(\boldsymbol{x_i}) \right) - \hat{\boldsymbol{x}}_{i,k} \right\|^2, \tag{1}$$

$$\text{s.t.} \quad \|\boldsymbol{x}_i - \boldsymbol{x}_j\| = L_j, \forall (i, j) \in \mathcal{B}. \tag{2}$$

where $H$ maps a inhomogeneous coordinate to equivalent homogeneous one. Note that $H^{-1}$ is the inverse process of $H$, not the function inverse:

$$\boldsymbol{y} \xrightarrow{H} \begin{bmatrix} \boldsymbol{y} \\ 1 \end{bmatrix}; \quad \begin{bmatrix} \boldsymbol{y} \\ w \end{bmatrix} \xrightarrow{H^{-1}} \frac{\boldsymbol{y}}{w} (w \neq 0). \tag{3}$$

### 3.2 Closed-Form Solution

**Reformulation of the Objective Function.** First, we analyze the objective function Eq. (1). Split the projection matrix by $\boldsymbol{P}_k = [\boldsymbol{P}_k^{u\top}, \boldsymbol{p}_k]^\top$ where $\boldsymbol{P}_k^u \in \mathbb{R}^{2\times4}, \boldsymbol{p}_k \in \mathbb{R}^4$. Then the objective function in Eq. (1) equals to

$$g(\boldsymbol{x}) = \sum_{i=0}^{n} \sum_{k=1}^{c} w'_{i,k} \left\| \boldsymbol{P}_k^u H(\boldsymbol{x}_i) - \hat{\boldsymbol{x}}_{i,k} \left( \boldsymbol{p}_k^\top H(\boldsymbol{x}_i) \right) \right\|^2. \tag{4}$$

where $\boldsymbol{x} = [\boldsymbol{x}_0^\top, \boldsymbol{x}_1^\top, \dots, \boldsymbol{x}_n^\top]^\top$ represents the full human pose, and the weight becomes $w'_{i,k} = w_{i,k} / \left( \boldsymbol{p}_k^\top H(\boldsymbol{x}_i) \right)^2$. We can ignore the term $\boldsymbol{p}_k^\top H(\boldsymbol{x}_i)$ and directly treat $w'_{i,k}$ as the new weight. Thus $g(\boldsymbol{x})$ becomes a quadratic function, whose minimization is trivial. Actually, if all weights are set the same, minimizing $g(\boldsymbol{x})$ will produce exactly the same result as linear-LS triangulation [13].

To better describe the constraints on bones in Eq. (2), we represent human pose with bones. Define a *bone vector* as a vector that points from proximal to distal joint of the bone. Use $\boldsymbol{b}_i$ to represent the bone vector with distal joint $i$ ($i = 1, 2, \dots, n$, no $\boldsymbol{b}_0$ because joint 0 is the root). Like $\boldsymbol{x}$, we concatenate all bone vectors to a single column vector $\boldsymbol{b} = [\boldsymbol{b}_1^\top, \boldsymbol{b}_2^\top, \dots, \boldsymbol{b}_n^\top]^\top$. Since $\boldsymbol{b}$ implies no global position, $\tilde{\boldsymbol{b}} = [\boldsymbol{x}_0^\top, \boldsymbol{b}^\top]$ is a comprehensive representation of human pose.

As is indicated in Kinematic Chain Space (KCS) [31], the conversion between joint positions and bone vectors is a linear process, and can be accomplished by matrix multiplication. Here, the matrix is defined as $\mathcal{G} = \{\mathcal{G}_{ij}\}$, where

$$\mathcal{G}_{ij} = \begin{cases} \boldsymbol{I}_3, & \text{if } i = j \text{ or joint } j-1 \text{ is the parent of } i-1; \\ -\boldsymbol{I}_3, & \text{if joint } j-1 \text{ is the child of } i-1; \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

Note that no bone vector can be linearly represented by the others, so all row vectors of $\mathcal{G}$ are linearly independent and $\mathcal{G}$ is non-singular. Then we have

$$\tilde{\boldsymbol{b}} = \mathcal{G}\boldsymbol{x}; \quad \boldsymbol{x} = \mathcal{G}^{-1}\tilde{\boldsymbol{b}}. \tag{6}$$

The conversion by Eq. (6) is not thorough because in $\tilde{\boldsymbol{b}}$, $\boldsymbol{x}_0$ is the root joint coordinate, not a bone vector. The constraints set no direct limit on $\boldsymbol{x}_0$, so we can fix $\boldsymbol{b}$ and solve $\boldsymbol{x}_0$ from an unconstrained quadratic optimization problem, which has a trivial solution. The optimal $\boldsymbol{x}_0$ is obtained via the following equation:

$$\boldsymbol{x}_0 = \boldsymbol{Q}\boldsymbol{b} + \boldsymbol{p} \tag{7}$$

where $\boldsymbol{Q} \in \mathbb{R}^{3\times 3n}, \boldsymbol{p} \in \mathbb{R}^3$ are known constants, whose expressions are provided in Sec. 1.1 Supplementary. We illustrate the full converting process from joint coordinates to bone vectors, as well as the inverse process, in Fig. 2.

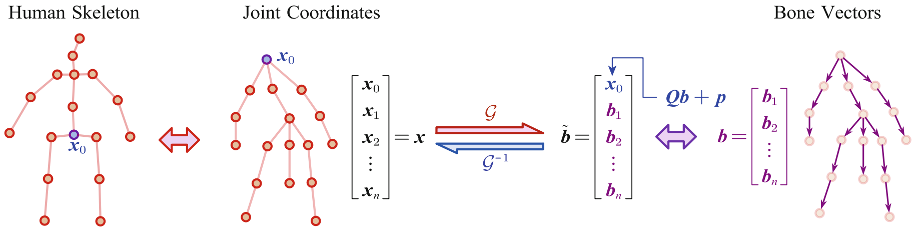

Human Skeleton
$x_0$
Joint Coordinates
$x_0$
$\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = x$
$\mathcal{G}$
$\mathcal{G}^{-1}$
$\tilde{b} = \begin{bmatrix} x_0 \\ b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$
$Qb + p$
$b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$
Bone Vectors

**Fig. 2.** The conversion between joint coordinates and bone vectors. From left to right, the human skeleton is represented by joint positions $\boldsymbol{x}$, which is further converted to $\tilde{\boldsymbol{b}}$ by Eq. (6). Then the root joint coordinate $\boldsymbol{x}_0$ is expressed by bone vectors and the conversion to $\boldsymbol{b}$ is done. The process is fully invertible.

With Eq. (7), Eq. (4) can be formulated as a quadratic function of $\boldsymbol{b}$. To help description, we mark the line as $l_{i,k}$, which connects the optic center of camera $k$ and the 2D estimation of joint $i$ on the image plane. Then we derive a property of the formulation of Eq. (4) and leave its proof in Sec. 1.2 in Supplementary.

*Property 1.* The optimization problem in Eq. (1) and (2) is formulated as

$$\begin{aligned} \min_{\boldsymbol{b}} \quad & f(\boldsymbol{b}) = \frac{1}{2}\boldsymbol{b}^\top \boldsymbol{A}\boldsymbol{b} - \boldsymbol{\beta}^\top \boldsymbol{b} + d && (8) \\ \text{s.t.} \quad & h_i(\boldsymbol{b}_i) = \|\boldsymbol{b}_i\|^2 = L_i^2,\ i = 1, 2, \ldots, n. && (9) \end{aligned}$$

where $\boldsymbol{A} \in \mathbb{R}^{3n \times 3n}$ is a symmetric positive semi-definite constant matrix, $\boldsymbol{\beta} \in \mathbb{R}^{3n}$ and $d \in \mathbb{R}$ are constants. $\boldsymbol{A}$ is singular if and only if $\exists i = 0, 1, \ldots, n$, there holds $\forall k_1, k_2 = 1, 2, \ldots, c, l_{i,k_1} /\!\!/ l_{i,k_2}$.

Actually when $\boldsymbol{A}$ is singular, excluding the factor of 2D estimation errors, all camera optic centers have to be approximately collinear with one of the joints. If we set up the cameras properly, such conditions can be easily avoided.

**Approximation and Linearization.** In Eq. (8), $f(\boldsymbol{b})$ is a convex function, but due to non-affine constraint in Eq. (9), the optimization problem is non-convex. Generally, we need to consider the necessary condition, i.e., Karush-Kuhn-Tucker (KKT) condition. The condition produces a nonlinear equation group, but we show that it is solvable under proper approximations.

Define three vectors: $\boldsymbol{h}(\boldsymbol{b}) = [h_1(\boldsymbol{b}_1), h_2(\boldsymbol{b}_2), \ldots, h_n(\boldsymbol{b}_n)]^\top$ represents the function that calculates square bone length vector given $\boldsymbol{b}$, $\boldsymbol{L} = [L_1^2, L_2^2, \ldots, L_n^2]^\top$ is the target square bone length vector, and $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_n]^\top$ is the multiplier vector. Then the Lagrange multiplier is written as:

$$l(\boldsymbol{b}, \boldsymbol{\lambda}) = f(\boldsymbol{b}) + \boldsymbol{\lambda}^\top \left(\boldsymbol{h}(\boldsymbol{b}) - \boldsymbol{L}\right). \tag{10}$$

The KKT condition gives us an equation group purely about $\boldsymbol{\lambda}$.

$$\boldsymbol{h}\left((\boldsymbol{A} + 2\boldsymbol{\Lambda})^{-1}\boldsymbol{\beta}\right) = \boldsymbol{L}, \tag{11}$$

where $\boldsymbol{\Lambda} = \operatorname{diag}\{\boldsymbol{\lambda}\} \otimes \boldsymbol{I}_3$ and "$\otimes$" is the Kronecker product. The unknown $\boldsymbol{\lambda}$ is inside a matrix inverse, which is highly non-linear. But we can make linear approximations about $(\boldsymbol{A} + 2\boldsymbol{\Lambda})^{-1}$ under certain assumptions. To support this, we introduce the following lemma (proof see Sec. 1.3 in Supplementary):

**Lemma 1.** *Suppose $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{n \times n}$ and $\boldsymbol{A}$ is non-singular. $\|\cdot\|$ is the spectral norm of a matrix. If $\|\boldsymbol{A}^{-1}\boldsymbol{B}\| < 1$, then $\boldsymbol{A} - \boldsymbol{B}$ is non-singular and the following inequality holds:*

$$\left\|(\boldsymbol{A} - \boldsymbol{B})^{-1} - \left(\boldsymbol{A}^{-1} + \boldsymbol{A}^{-1}\boldsymbol{B}\boldsymbol{A}^{-1}\right)\right\| \leq \frac{\left\|\boldsymbol{A}^{-1}\boldsymbol{B}\right\|^2 \left\|\boldsymbol{A}^{-1}\right\|}{1 - \left\|\boldsymbol{A}^{-1}\boldsymbol{B}\right\|}. \tag{12}$$

By substituting $-2\boldsymbol{\Lambda}$ for $\boldsymbol{B}$ in Lemma 1, we directly conclude that

$$(\boldsymbol{A} + 2\boldsymbol{\Lambda})^{-1} \approx \boldsymbol{A}^{-1} - 2\boldsymbol{A}^{-1}\boldsymbol{\Lambda}\boldsymbol{A}^{-1}. \tag{13}$$

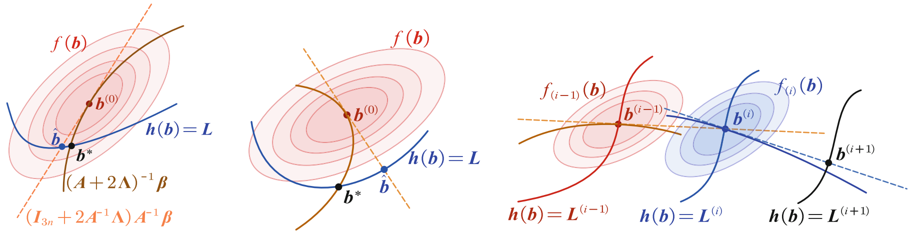
Equation (13) provides a linear approximation for matrix inverse. After replacing $(\boldsymbol{A} + 2\boldsymbol{\Lambda})^{-1}$ in Eq. (11), there is still a second-order term of $\boldsymbol{\lambda}$. We can abandon it with respect to $\left\|2\boldsymbol{A}^{-1}\boldsymbol{\varLambda}\right\| \ll 1$ so that Eq. (11) becomes totally linear. Define some notations. $\boldsymbol{L}^{(i)} = \left[\|\boldsymbol{b}_1^{(i)}\|^2, \|\boldsymbol{b}_2^{(i)}\|^2, \cdots, \|\boldsymbol{b}_n^{(i)}\|^2\right]^\top$, $\boldsymbol{b}^{(0)} = \boldsymbol{A}^{-1}\boldsymbol{\beta}$ is the minimizer of $f(\boldsymbol{b})$ with no constraint, which is later referred to as the *initial solution*, and $\boldsymbol{D}_n^{(3\times1)} \in \mathbb{R}^{3n\times n}$ represents a block diagonal matrix whose diagon is filled by $n$ 3-dimensional all-1 column vectors. Then the expression of $\boldsymbol{\lambda}$ is

$$\boldsymbol{\lambda} = \frac{1}{4}\left(\boldsymbol{D}_n^{(3\times1)\top}\mathrm{diag}\{\boldsymbol{b}^{(0)}\}\boldsymbol{A}^{-1}\mathrm{diag}\{\boldsymbol{b}^{(0)}\}\boldsymbol{D}_n^{(3\times1)}\right)^{-1}(\boldsymbol{L}^{(0)} - \boldsymbol{L}). \tag{14}$$

The bone vector estimation is given by

$$\hat{\boldsymbol{b}} = \boldsymbol{b}^{(0)} - 2\boldsymbol{A}^{-1}\boldsymbol{\varLambda}\boldsymbol{b}^{(0)}, \tag{15}$$

With Eq. (15), we can derive root joint coordinate from $\hat{\boldsymbol{b}}$ by Eq. (7), and then applying Eq. (6) reversely will produce the optimal pose. This is the theory of Structural Triangulation. The process is illustrated more clearly in Fig. 3a.


$f(\boldsymbol{b})$
$\boldsymbol{b}^{(0)}$
$\hat{\boldsymbol{b}}$
$\boldsymbol{b}^*$
$\boldsymbol{h}(\boldsymbol{b}) = \boldsymbol{L}$
$(\boldsymbol{A}+2\boldsymbol{\Lambda})^{-1}\boldsymbol{\beta}$
$(\boldsymbol{I}_{3n}+2\boldsymbol{A}^{-1}\boldsymbol{\Lambda})\boldsymbol{A}^{-1}\boldsymbol{\beta}$
$f(\boldsymbol{b})$
$\boldsymbol{b}^{(0)}$
$\boldsymbol{h}(\boldsymbol{b}) = \boldsymbol{L}$
$\boldsymbol{b}^*$
$\hat{\boldsymbol{b}}$
$f_{(i-1)}(\boldsymbol{b})$
$\boldsymbol{b}^{(i-1)}$
$f_{(i)}(\boldsymbol{b})$
$\boldsymbol{b}^{(i)}$
$\boldsymbol{b}^{(i+1)}$
$\boldsymbol{h}(\boldsymbol{b}) = \boldsymbol{L}^{(i-1)}$
$\boldsymbol{h}(\boldsymbol{b}) = \boldsymbol{L}^{(i)}$
$\boldsymbol{h}(\boldsymbol{b}) = \boldsymbol{L}^{(i+1)}$

(a) Linearize process (b) Diverge case (c) SCA update process

**Fig. 3.** Illustrations of solution finding process. Groups of ellipses represent the level sets of objective functions. In (a) and (b), blue curve is the bone length constraint, brown line is the range of $(\boldsymbol{A} + \boldsymbol{\varLambda})^{-1}\boldsymbol{\beta}$ as $\boldsymbol{\lambda}$ varies, and orange virtual line is the linearized range by Eq. (13). $\boldsymbol{b}^*$ is the analytical solution while $\hat{\boldsymbol{b}}$ is the approximated one. They are usually close, as (a) shows, but sometimes not, like (b). For diverge cases, SCA will be helpful. Figure (c) shows the update process, where we mark variables at adjacent stages with different colors. From step $i-1$ to $i$, the solution is updated to $\boldsymbol{b}^{(i)}$ in the way shown in (a) and a new quadratic objective function is formed with $\boldsymbol{b}^{(i)}$ as the minimizer. Then we target for the bone lengths constraint, i.e., $\boldsymbol{h}(\boldsymbol{b}) = \boldsymbol{L}^{(i+1)}$ and repeat the process. (Color figure online)

### 3.3 SCA: Step Constraint Algorithm

The above analysis is under the assumption of $\left\|2\boldsymbol{A}^{-1}\boldsymbol{\varLambda}\right\| \ll 1$, but sometimes it fails to hold. Structural Triangulation may be regarded as a process to correct a pose so that the squared bone length vector changes from $\boldsymbol{L}^{(0)}$ to the target

**Algorithm 1:** Structural Triangulation + Step Constraint Algorithm

```
   Input  : $\boldsymbol{A}, \boldsymbol{\beta}, \boldsymbol{L}, N, \alpha_0, \alpha_1, \ldots, \alpha_N$
   Output: $\boldsymbol{b}$
1  $\boldsymbol{T}^{(0)} \longleftarrow \boldsymbol{A}^{-1}$;
2  $\boldsymbol{b}^{(0)} \longleftarrow \boldsymbol{T}^{(0)}\boldsymbol{\beta}$;
3  for $i \leftarrow 1$ to $N$ do
4      $\boldsymbol{L}_{real}^{(i-1)} \longleftarrow \boldsymbol{h}(\boldsymbol{b}^{(i-1)})$; // when i=1, initialize L0.
5      $\boldsymbol{L}^{(i)} \longleftarrow \left(\alpha_i \boldsymbol{L}_{real}^{(i-1)} + (\alpha_{i-1} - \alpha_i)\boldsymbol{L}\right)/\alpha_{i-1}$;
6      $\boldsymbol{\lambda} \longleftarrow \left(\boldsymbol{D}_n^{(3\times1)\top} \mathrm{diag}\{\boldsymbol{b}^{(i-1)}\}\boldsymbol{T}^{(0)}\mathrm{diag}\{\boldsymbol{b}^{(i-1)}\}\boldsymbol{D}_n^{(3\times1)}\right)^{-1}(\boldsymbol{L}_{real}^{(i-1)} - \boldsymbol{L}^{(i)})/4$;
7      $\boldsymbol{\Lambda} \longleftarrow \mathrm{diag}\{\boldsymbol{\lambda}\} \otimes \boldsymbol{I}_3$;
8      $\boldsymbol{T}^{(i)} \longleftarrow (\boldsymbol{I}_n - 2\boldsymbol{T}^{(i-1)}\boldsymbol{\Lambda})\boldsymbol{T}^{(i-1)}$;
9      $\boldsymbol{b}^{(i)} \longleftarrow \boldsymbol{T}^{(i)}\boldsymbol{\beta}$;
10 end
11 $\boldsymbol{b} \longleftarrow \boldsymbol{b}^{(N)}$;
```

$\boldsymbol{L}$. In Eq. (14), it is obvious that if the initial bone lengths are far from target, which is possible when 2D estimations are not precise enough, then $\|\boldsymbol{\lambda}\|$ can be large and the assumption may be contradicted. Consequently, the result may diverge (as shown in Fig. 3b).

However, this case is still in reach with some modifications. We can interpolate some points between $\boldsymbol{L}^{(0)}$ and $\boldsymbol{L}$ and use ST to correct pose from one point to the next. Adjacent points are near so the assumption is confirmed to hold each time. This is the basic idea of *Step Constraint Algorithm*.

First we need to determine a *step number* $N$ and $N-1$ step points between $\boldsymbol{L}^{(0)}$ and $\boldsymbol{L}$, which are marked as $\boldsymbol{L}^{(1)}, \boldsymbol{L}^{(2)}, \ldots, \boldsymbol{L}^{(N-1)}$. Let $\boldsymbol{L}^{(N)} = \boldsymbol{L}$. A way to generate them is by constructing a decreasing series $\{\alpha_i\}_{i=0}^{N}$ ($\alpha_0 = 1, \alpha_N = 0$), and calculate by linear interpolating, where the series serve as the proportions, i.e., $\boldsymbol{L}^{(i)} = \alpha_i \boldsymbol{L}^{(0)} + (1-\alpha_i)\boldsymbol{L}$. However, in iteration from $i-1$ to $i$, $\boldsymbol{h}(\boldsymbol{b}^{(i)})$ does not exactly equal to the predefined splitting point $\boldsymbol{L}^{(i)}$ due to approximations. So a better way is to determine these splitting points in-the-run. In other words, we first calculate the real square bone length vector $\boldsymbol{L}_{real}^{(i-1)} = \boldsymbol{h}(\boldsymbol{b}^{(i-1)})$, then find $\boldsymbol{L}^{(i)}$ by $(\boldsymbol{L}^{(i)} - \boldsymbol{L})/\alpha_i = (\boldsymbol{L}^{(i-1)} - \boldsymbol{L})/\alpha_{i-1}$.

$$\boldsymbol{L}^{(i)} = \frac{\alpha_i}{\alpha_{i-1}}\boldsymbol{L}_{real}^{(i-1)} + \frac{\alpha_{i-1} - \alpha_i}{\alpha_{i-1}}\boldsymbol{L} \tag{16}$$

In our experiment, we use the in-the-run method and determine $\{\alpha_i\}_{i=0}^{N}$ simply by $\alpha_i = (N-i)/N$. In experiment, if not specified, then $N$ is set to 3.

The pseudo-code of ST + SCA is shown in Algorithm 1. Besides $\boldsymbol{b}^{(i)}$ and $\boldsymbol{L}^{(i)}$, $\boldsymbol{T}^{(i)}$ represents another important variable to update - the approximation of $(\boldsymbol{A} + 2\boldsymbol{\Lambda})^{-1}$. The update is done in line 8 of Algorithm 1, by the linear expression provided in Eq. (13). Figure 3c illustrates the whole process.

# 4 Experiments

## 4.1 Experimental Settings

**Datasets and Metrics.** In the experiments, we use two public datasets: Human3.6M [15] and Total Capture [29] datasets.

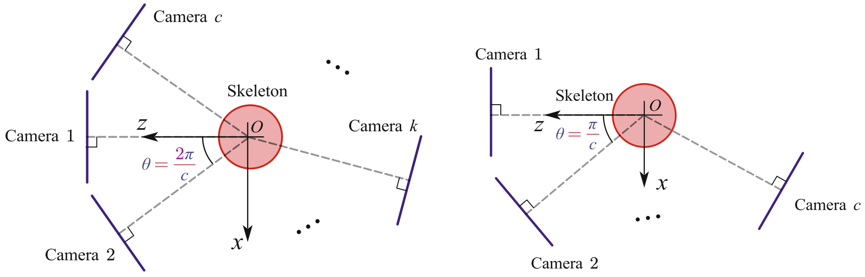
In Human3.6M Dataset, the images are acquired by 4 cameras at 50 Hz and the dataset contains more than 3.6 million images, which are organized by different subjects. By convention, S1, S5, S6, S7, and S8 are used for training, while S9 and S11 are used for testing. Note that Human3.6M provides pose labels in 32-joint form, and we follow the common criterion to use the 17-joint subset.

In Total Capture Dataset, 8 cameras are used to capture images, where we use cameras 1, 3, 5, and 7. The data are also organized by subjects. The test set contains “Walking-2”(W2), “Freestyle-3”(FS3), and “Acting-3” (A3) of all 5 subjects. Note that the original labels are arranged in 21-joint form, with which “Nose” joint in previous 17-joint model fails to make correspondence. So we use a 16-joint subset (details see Sec. 2 in Supplementary) for ST, where “RightArm” and “Neck”, “LeftArm” and “Neck” are not directly connected in the original skeletal model and the lengths may vary in a limited range.

For metrics, we use Mean Per Joint Position Error (MPJPE) to measure joint precision, along with some new metrics on bone lengths. Two types of MPJPEs are used: (1) absolute MPJPE (MPJPE-ab) calculate the position errors directly without alignment; (2) relative MPJPE (MPJPE-re), usually known as Protocol #1 [22], measures position errors after aligning the pelvis. Since labels of some subsets in S9 is shifted, we follow [16] to present MPJPE-ab after eliminating these bad samples. Additionally, we introduce some metrics on bones: (1) Mean Per Bone Length Error (MPBLE) measures the average over all bone length errors, the same as MPLLE in [21]. (2) Mean Bone Length Standard deviation (MBLS) equals the square root of average variance over all bone lengths. (3) Percentage of Inlier Bones (PIB) is the rate of bones with reasonable lengths, to be exact, 0.8˜1.2 times the true bone lengths.

**The Choice of 2D Estimation Model.** Because 2D HPE is not involved in our work, a proper 2D backbone is needed to test our method on public datasets. We choose the algebraic triangulation model by Iskakov et al. [16] because it is a precise and simple framework. The model, which serves as our *baseline*, consists of a 2D backbone and a SVD triangulation module. It also provides beliefs for cameras which we use as weights in Eq. (4). In our experiment, the model is pretrained on MPII [1] and fine-tuned on Human3.6M dataset. We keep the 2D backbone and replace the triangulation module with our method.

**Virtual Test Settings.** We aim to prove that our method outperforms conventional triangulation methods once the bone lengths are known, regardless of camera settings and 2D backbones. So we synthesize 2D estimations randomly by modeling the 2D estimations as “ground truth re-projection + Gaussian noise”.



(a) Round (b) Half

**Fig. 4.** Two patterns to set up virtual cameras, where $c$ is the camera number and $\theta$ is the angle interval between adjacent camera optic axes. $O$ is the global coordinate origin, which all skeleton root joints are aligned to. All image planes are orthogonal to the principal optic axes, while these axes all point to $O$ and distribute uniformly in predefined angle range, i.e., $2\pi$ in (a) and $\pi$ in (b). The axis of camera 1 is right on $z$ axis.

After generating $c$ virtual cameras with projection matrices $\boldsymbol{P}_1, \boldsymbol{P}_2, \ldots, \boldsymbol{P}_c$, we can re-project 3D pose and get the ground truth 2D poses. Concatenate these coordinates as a column vector $\boldsymbol{x}_{2d} \in \mathbb{R}^{2(n+1)}$. Then generate a noise vector $\boldsymbol{\epsilon} \in \mathbb{R}^{2(n+1)}$, each of whose element obeys Gaussian distribution $N(0, \sigma)$. Finally the generated 2D estimations are calculated by $\hat{\boldsymbol{x}}_{2d} = \boldsymbol{x}_{2d} + \boldsymbol{\epsilon}$.

In our experiment, all cameras share the same intrinsics, along with two types of camera extrinsic settings: round and half (Fig. 4). We use ground truth labels from sampled test set of Human3.6M Dataset - totally 2181 frames - as our base 3D pose to confirm feasibility and variety.

### 4.2 Experiments on Public Datasets

**Quantitative Results and Analysis.** The test result of two datasets are reported in Table 1, 2, and 3.

The acquisition of bone lengths is simple in public datasets since we can use the ground truth of one frame to calculate. In practice, it is also available by mature human measurement techniques. However, we need to consider the errors in bone length measurements. We therefore provide a simple estimation by averaging all symmetric bone lengths in linear triangulation results of T-pose frames. We mark experiments using such bone lengths with "*".

Table 1 and 2 show the experiment result on Human3.6M dataset. Besides linear triangulation, we also implement an iterative optimization algorithm - Lagrangian algorithm [7] - to solve the problem described by Eq. (8) and (9) to serve as a baseline (details see Sec. 3.1 in Supplementary). As is shown, our method exceeds the previous state-of-the-art method (Volumetric Triangulation [16]) by 4.9% in MPJPE-re. Absolute MPJPE error is also reduced by 0.15 mm. We can also observe that SCA helps lower MPJPE-re by 0.87 mm and MPJPE-

**Table 1.** Relative MPJPE (mm) on Human3.6M Dataset compared with previous state-of-the-art methods. We highlight tests in our method in light gray, and "*" means estimated bone lengths are used.

| Method | Dire. | Disc. | Eat | Greet | Phone | Photo | Pose | Purch. |
|---|---|---|---|---|---|---|---|---|
| Tome et al. [28] | 43.3 | 49.6 | 42.0 | 48.8 | 51.1 | 64.3 | 40.3 | 43.3 |
| Yihui and Rui et al.[14] | 28.9 | 32.5 | 26.6 | 28.1 | 28.3 | 29.3 | 28.0 | 36.8 |
| Remelli et al. [25] | 27.3 | 32.1 | 25.0 | 26.5 | 29.3 | 35.4 | 28.8 | 31.6 |
| Qiu et al. [24] | 23.98 | 26.71 | 23.19 | 24.30 | 24.77 | 22.82 | 24.12 | 28.62 |
| AT by Iskakov et al. [16] | 20.42 | 22.83 | 19.98 | 19.48 | 21.73 | 20.69 | 19.11 | 22.39 |
| VT by Iskakov et al. [16] | 18.06 | 19.63 | 19.45 | 18.36 | 19.95 | 19.36 | 17.79 | 20.68 |
| Lagrangian algorithm [7] | 19.33 | 20.85 | 18.68 | 18.49 | 21.77 | 20.05 | 17.97 | 20.89 |
| Ours* | 18.67 | 21.27 | 17.95 | 18.90 | 21.00 | 19.18 | 18.48 | 22.07 |
| Ours (w/o SCA) | 17.56 | 20.03 | 16.22 | 17.86 | 20.49 | 19.06 | 17.38 | 22.08 |
| Ours | **17.37** | **19.70** | **15.56** | **17.46** | **19.61** | **18.82** | **16.95** | **20.24** |
| Method | Sit | SitD | Smoke | Wait | Walk | WalkD | WalkT | Avg. |
| Tome et al. [28] | 66.0 | 95.2 | 50.2 | 52.2 | 51.1 | 43.9 | 45.3 | 52.8 |
| Yihui and Rui et al. [14] | 42.0 | 30.5 | 35.6 | 30.0 | 29.3 | 30.0 | 30.5 | 31.2 |
| Remelli et al. [25] | 36.4 | 31.7 | 31.2 | 29.9 | 26.9 | 33.7 | 30.4 | 30.2 |
| Qiu et al. [24] | 32.12 | 26.87 | 30.98 | 25.56 | 25.02 | 28.07 | 24.37 | 26.21 |
| AT by Iskakov et al. [16] | 26.10 | 31.80 | 22.85 | 20.94 | 20.13 | 23.50 | 21.12 | 22.33 |
| VT by Iskakov et al. [16] | 23.27 | 29.43 | 20.58 | 19.38 | 18.66 | 21.15 | 19.12 | 20.35 |
| Lagrangian algorithm [7] | 24.99 | 29.18 | 21.89 | 19.94 | 19.37 | 22.05 | 20.28 | 21.18 |
| Ours* | 23.34 | 28.17 | 20.73 | 20.27 | 20.25 | 21.87 | 20.19 | 20.86 |
| Ours (w/o SCA) | 24.07 | 29.87 | 20.44 | 19.83 | 17.96 | 20.97 | 18.91 | 20.22 |
| Ours | **21.92** | **26.71** | **19.25** | **18.90** | **17.88** | **20.64** | **18.69** | **19.35** |

**Table 2.** Average absolute MPJPE (mm) on Human3.6M Dataset.

| Method | MPJPE-ab |
|---|---|
| Baseline | 19.26 |
| Vol. [16] | 17.93 |
| Ours* | 18.90 |
| Ours (w/o SCA) | 17.98 |
| Ours | **17.78** |

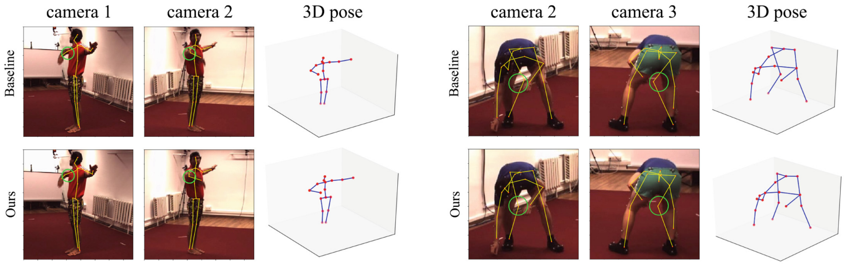
**Table 3.** Relative MPJPE (mm) on Total Capture Dataset. The bone lengths used in tests of the last line are the average over all ground truth labels instead of one frame.

| Method | W2 | FS3 | A3 | Average |
|---|---|---|---|---|
| Baseline | **69.0** | 65.8 | 56.0 | 63.3 |
| Ours* | 73.0 | 70.2 | 58.4 | 66.9 |
| Ours | 69.2 | **60.2** | **50.2** | **59.6** |

ab by 0.20 mm. Our method outperforms the iterative baseline, and reaches satisfying accuracy when imprecise bone lengths are used.

We also study how the step number $N$ in SCA affects precision. Relatie MPJPE generally decreases with $N$ and gets 19.33 mm at $N = 9$. However, when $N \geq 10$ the estimations in some frames will diverge, causing an abnormal error increase. So a relatively small number is recommended, like 3 in our tests.

In experiments on Total Capture Dataset, we focus on whether our method works in case some connection lengths are not actually fixed. We report the result in Table 3. Though our method does not correct 3D pose successfully when bone length estimations are imprecise, a 3.7 mm decrease in MPJPE-re error is obtained when bone lengths are known.



(a) An example in S9 (b) An example in S11

**Fig. 5.** Examples on how Structural Triangulation corrects human pose. The upper row is the result by SVD triangulation while the lower is by ours. Only 2 representative views are selected from the 4 views for illustration. The circled point in (a) gets 61.7% reduction in relative joint position error, while in (b) the reduction is 24.3%.

**Qualitative Analysis.** To describe how our method corrects poses, we take two examples from test subjects in Human3.6M dataset and mark the remarkably improved points with circles in Fig. 5. The initial pose possesses shorter right arms in Fig. 5a and longer right leg in Fig. 5b. We can see how the pose is corrected while pursuing the correct bone lengths.

**Ablation Study.** In this section, our major concern is the improvement on bone lengths of our method, and the affect of whether precise bone lengths are given. The metrics on bones proposed in Sect. 4.1 are used. We report the result in Table 4. The use of ground truth bone lengths is treated as a component to help analyze the effect of imprecise bone length input.

In the first two rows, it is clear that merely ST is enough to decrease MPBLE by over 35% in S9 and S11 even with estimated bone lengths. It implies the increase in bone length precision, yet we still need to study MBLS and PIB to conclude the reason. Actually, smaller MBLS means stabler bone lengths in estimation, and larger PIB indicates larger proportion of reasonable poses.

Compared to baseline, ST has the effect to stabilize bone lengths, and SCA makes the effect even stronger. In the last two rows, small MBLS and 100% PIB indicate that the bone lengths are nearly invariant, which proves that our method constrains the bone lengths in a strict way.

**Table 4.** Effects of whether SCA is involved ("SCA") or whether ground truth bone lengths are used ("GT"). The unit for MPBLE and MBLS is mm. PIB is presented in percentage (%). Down arrows mean the smaller the better while up arrows mean the contrary. "*" means estimated bone lengths are used.

| Method | SCA | GT | S9 | | | S11 | | |
|---|---|---|---|---|---|---|---|---|
| | | | MPBLE ↓ | MBLS ↓ | PIB ↑ | MPBLE ↓ | MBLS ↓ | PIB ↑ |
| Baseline | – | – | 12.2 | 15.9 | 97.4 | 7.62 | 9.88 | 97.5 |
| ST* | × | × | 7.93 | 0.350 | 99.9 | 5.48 | 0.156 | **100** |
| ST | × | √ | 1.31 | 4.35 | 99.8 | 0.531 | 1.19 | **100** |
| ST + SCA* | √ | × | 7.93 | **0.122** | **100** | 5.48 | 0.0243 | **100** |
| ST + SCA | √ | √ | **0.129** | 0.151 | **100** | **0.0555** | **0.0178** | **100** |

### 4.3 Experiments on Synthesized 2D Estimations

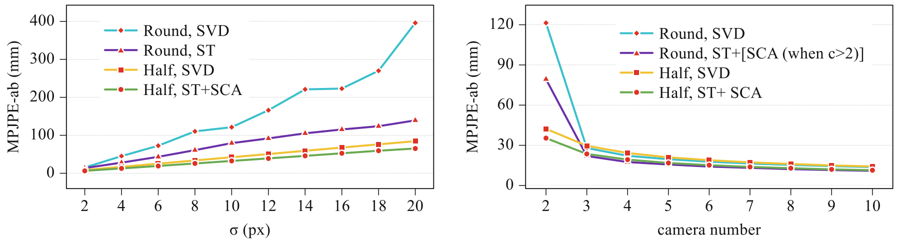
We conduct experiments on data generated in the way proposed in Sect. 4.1, where noise standard deviation $\sigma$ varies from 2 px to 20 px, and camera number $c$ varies from 2 to 10 and all combinations are considered. Some representative results are plotted in Fig. 6 and the full results are available in Sec. 3.4 in Supplementary.

Clearly, our method shows better precision than SVD triangulation. In Fig. 6, we observe boundary effect in the promotion of ST, but ST has certain positive effect in all experiment settings. We also calculate the proportion of frames when ST outperforms the baseline under all combinations of $\sigma$ and $c$ which is always more than 82% and nearly 100% when there are more than 2 cameras. In conclusion, the generality of our method is validated.

### 4.4 Running Time

Now we would like to validate the computational efficiency. We conduct experiments to compare our method with Volumetric Triangulation Model [16], RPSM [24], and SVD Triangulation method. Since [16] does not generate 2D estimations but require algobraic model to generate a rough estimation, the time is how much the whole end-to-end process takes. We run different methods on the same computer with a 16-core 2.10 GHz Intel E5-2620 v4 CPU, an Nvidia Titan Xp GPU, 32 GB RAM. The experiment results are reported in Table 5.

As shown in Table 5, our method is much faster than RPSM and Volumetric Triangulation. Additionally, more steps in SCA is not costing much time. Though it takes more time than basic SVD triangulation, compared to time cost in 2D backbones (about 400 ms) in our test, the increase is not obvious.



(a) MPJPE-$\sigma$ curve

(b) MPJPE-$c$ curve

**Fig. 6.** Curves on how absolute MPJPE varies with the change of noise standard deviation $\sigma$ and camera number $c$. In (a) we set $c = 2$, while in (b) we set $\sigma = 10$ px. When there are only 2 cameras in round camera setting, they are right on the opposite directions to each other, which causes singularity in SCA. So in such experiments we eliminate SCA step and apply pure ST.

**Table 5.** Per frame inference time of different methods. The numbers of steps is the values of $N$ used in SCA.

| Method | SVD | RPSM [24] | Vol. [16] | Ours (3 steps) | Ours (9 steps) |
|---|---|---|---|---|---|
| Inference time (ms) | 1.95 | $1.82 \times 10^3$ | 305.4 | 6.63 | 8.96 |

## 5 Conclusions

In this paper we formulate the problem of 2D-to-3D inference in multi-view 3D HPE as a constrained optimization problem, and propose a novel closed-form solution, i.e., Structural Triangulation. To further generalize our method, we design SCA to make it compatible with the situation when large error occurs in 2D estimations. Experiments on open datasets and synthesized data prove our method is effective, generally applicable, and efficient.

## References

1. Andriluka, M., Pishchulin, L., Gehler, P., Schiele, B.: 2D human pose estimation: new benchmark and state of the art analysis. In: CVPR (2014)
2. Belagiannis, V., Amin, S., Andriluka, M., Schiele, B., Navab, N., Ilic, S.: 3D pictorial structures for multiple human pose estimation. In: CVPR (2014)
3. Bogo, F., et al.: Keep it SMPL: automatic estimation of 3D human pose and shape from a single image. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9909, pp. 561–578. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46454-1_34
4. Burenius, M., Sullivan, J., Carlsson, S.: 3D pictorial structures for multiple view articulated pose estimation. In: CVPR (2013)
5. Chen, H., Guo, P., Li, P., Lee, G.H., Chirikjian, G.: Multi-person 3D pose estimation in crowded scenes based on multi-view geometry. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12348, pp. 541–557. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58580-8_32

6. Chen, L., Ai, H., Chen, R., Zhuang, Z., Liu, S.: Cross-view tracking for multi-human 3D pose estimation at over 100 fps. In: CVPR (2020)
7. Chong, E.K., Zak, S.H.: An Introduction to Optimization. John Wiley, Hoboken (2004)
8. Conn, A.R., Gould, N.I., Toint, P.L.: Trust Region Methods. SIAM, Philadelphia (2000)
9. Dabral, R., Mundhada, A., Kusupati, U., Afaque, S., Sharma, A., Jain, A.: Learning 3D human pose from structure and motion. In: ECCV (2018)
10. Dong, Z., Song, J., Chen, X., Guo, C., Hilliges, O.: Shape-aware multi-person pose estimation from multi-view images. In: ICCV (2021)
11. Fischler, M., Elschlager, R.: The representation and matching of pictorial structures. IEEE Trans. Comput. C- **22**(1), 67–92 (1973). https://doi.org/10.1109/T-C.1973.223602
12. Gall, J., Stoll, C., de Aguiar, E., Theobalt, C., Rosenhahn, B., Seidel, H.P.: Motion capture using joint skeleton tracking and surface estimation. In: CVPR (2009)
13. Hartley, R.I., Sturm, P.: Triangulation. CVIU **68**(2), 146–157 (1997). https://doi.org/10.1006/cviu.1997.0547, http://www.sciencedirect.com/science/article/pii/S1077314297905476
14. He, Y., Yan, R., Fragkiadaki, K., Yu, S.I.: Epipolar transformers. In: CVPR (2020)
15. Ionescu, C., Papava, D., Olaru, V., Sminchisescu, C.: Human3.6m: large scale datasets and predictive methods for 3D human sensing in natural environments. TPAMI **36**(7), 1325–1339 (2014). https://doi.org/10.1109/TPAMI.2013.248
16. Iskakov, K., Burkov, E., Lempitsky, V., Malkov, Y.: Learnable triangulation of human pose. In: ICCV (2019)
17. Kolotouros, N., Pavlakos, G., Black, M.J., Daniilidis, K.: Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In: ICCV (2019)
18. Li, J., Xu, C., Chen, Z., Bian, S., Yang, L., Lu, C.: Hybrik: a hybrid analytical-neural inverse kinematics solution for 3D human pose and shape estimation. In: CVPR (2021)
19. Li, X., Fan, Z., Liu, Y., Li, Y., Dai, Q.: 3D pose detection of closely interactive humans using multi-view cameras. Sensors **19**(12), 2831 (2019). https://doi.org/10.3390/s19122831, https://www.mdpi.com/1424-8220/19/12/2831
20. Lin, J., Lee, G.H.: Multi-view multi-person 3D pose estimation with plane sweep stereo. In: CVPR (2021)
21. Ma, X., Su, J., Wang, C., Ci, H., Wang, Y.: Context modeling in 3D human pose estimation: a unified perspective. In: CVPR (2021)
22. Martinez, J., Hossain, R., Romero, J., Little, J.J.: A simple yet effective baseline for 3D human pose estimation. In: ICCV (2017)
23. Pavlakos, G., Zhou, X., Derpanis, K.G., Daniilidis, K.: Harvesting multiple views for marker-less 3D human pose annotations. In: CVPR (2017)
24. Qiu, H., Wang, C., Wang, J., Wang, N., Zeng, W.: Cross view fusion for 3D human pose estimation. In: ICCV (2019)
25. Remelli, E., Han, S., Honari, S., Fua, P., Wang, R.: Lightweight multi-view 3D pose estimation through camera-disentangled representation. In: CVPR (2020)
26. Rhodin, H., et al.: Learning monocular 3D human pose estimation from multi-view images. In: CVPR (2018)
27. Sharma, S., Varigonda, P.T., Bindal, P., Sharma, A., Jain, A.: Monocular 3D human pose estimation by generation and ordinal ranking. In: ICCV (2019)
28. Tome, D., Toso, M., Agapito, L., Russell, C.: Rethinking pose in 3D: multi-stage refinement and recovery for markerless motion capture. In: 3DV (2018)

29. Trumble, M., Gilbert, A., Malleson, C., Hilton, A., Collomosse, J.: Total capture: 3d human pose estimation fusing video and inertial sensors. In: BMCV (2017)
30. Tu, H., Wang, C., Zeng, W.: VoxelPose: towards multi-camera 3D human pose estimation in wild environment. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12346, pp. 197–212. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58452-8_12
31. Wandt, B., Ackermann, H., Rosenhahn, B.: A kinematic chain space for monocular motion capture. In: ECCV Workshops (2018)
32. Xie, R., Wang, C., Wang, Y.: Metafuse: a pre-trained fusion model for human pose estimation. In: CVPR (2020)
33. Xu, J., Yu, Z., Ni, B., Yang, J., Yang, X., Zhang, W.: Deep kinematics analysis for monocular 3D human pose estimation. In: CVPR (2020)
34. Yang, Y., Ramanan, D.: Articulated pose estimation with flexible mixtures-of-parts. In: CVPR (2011)
35. Yao, Y., Jafarian, Y., Park, H.S.: Monet: Multiview semi-supervised keypoint detection via epipolar divergence. In: ICCV (2019)
36. Zeng, A., Sun, X., Yang, L., Zhao, N., Liu, M., Xu, Q.: Learning skeletal graph neural networks for hard 3D pose estimation. In: ICCV (2021)