

Human Spine Motion Capture using Perforated Kinesiology Tape

Hendrik Hachmann

Bodo Rosenhahn

Institute for Information Processing (tnt) / L3S, Leibniz University Hannover

{hachmann, rosenhahn}@tnt.uni-hannover.de

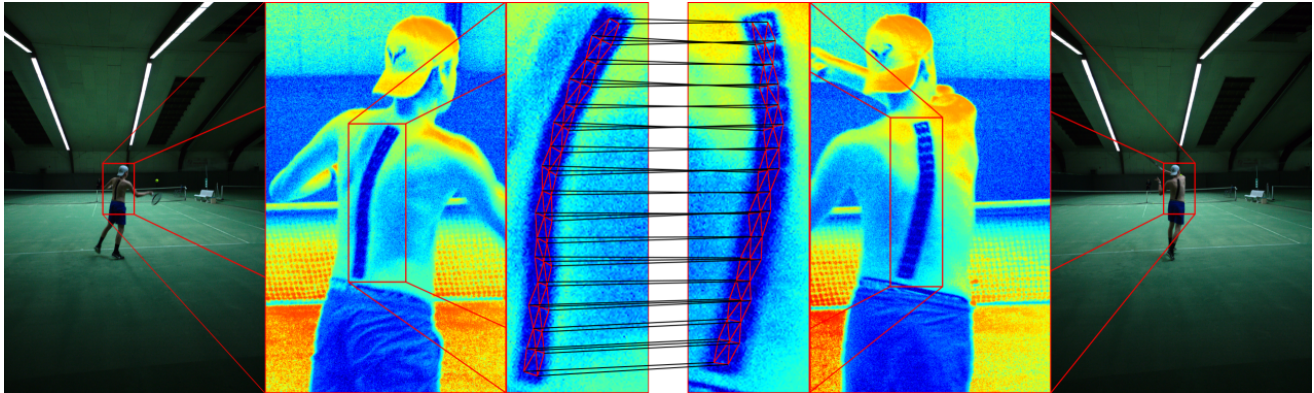


Figure 1. Spine tracking during a forehand in Tennis. Perforated kinesiology tape on the back of the player is used as a marker. The arrangement of its dots is known a priori and exploited for matching in a multi view 3D tracking system.

Abstract

In this work, we present a marker-based multi-view spine tracking method that is specifically adjusted to the requirements for movements in sports. A maximal focus is on the accurate detection of markers and fast usage of the system. For this task, we take advantage of the prior knowledge of the arrangement of dots in perforated kinesiology tape. We detect the tape and its dots using a Mask R-CNN and a blob detector. Here, we can focus on detection only while skipping any image-based feature encoding or matching. We conduct a reasoning in 3D by a linear program and Markov random fields, in which the structure of the kinesiology tape is modeled and the shape of the spine is optimized. In comparison to state-of-the-art systems, we demonstrate that our system achieves high precision and marker density, is robust against occlusions, and capable of capturing fast movements.

1. Introduction

Human motion capture is a long-studied field that is of particular interest in application fields like health, feature film production, and also increasingly in sports. It allows to capture the movements and techniques of athletes and

uncovers specific traits, characteristics, that may give them an advantage but also problems in the performance that might raise health problems. One motivation is straightforward: If the motion of the professional athlete of your choice is captured with sufficient precision, a motion capture system may tell you how to improve yourself to come close to your idol. This is one of several reasons for the widespread publicity in news and media accompanying the 2018 study of Schepers *et al.* [21], when they captured Tennis players at Wimbledon using an Xsens MVN Link suit. Their tracking system can be further used for extensive bio-mechanical analyses, i.e. that investigate how forces propagate through the body of an athlete in a musculoskeletal model as OpenSIM [6]. This may be used to design person specific training plans to achieve damage precaution and consequently, avoid injuries.

Back pain has become a major disease of civilization that affects large parts of the population. Some causes are widely known, such as lack of exercise or poor posture, e.g. when sitting at a desk at the workplace. But lack of back mobility and incorrect technique in sports might also lead to back pain. In a spinal kinematics study by Campbell *et al.* [5] with the 2014 Tennis Australia National Squad the authors found a connection between lower back pain and lumbar flexibility during serve kinematics.

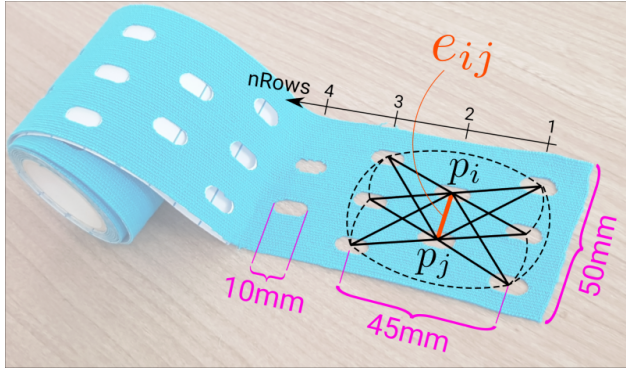


Figure 2. (background): A roll of kinesiology tape (foreground): Spatial dimensions as the mutual distance are exploited as 3D a priori knowledge of the structure of the tape. (graph): The two points p_i and p_j define the edge e_{ij} . All of the black edges are defined as the neighboring edges of e_{ij} and denoted as Ω_e .

The vertebral column consists of 24 vertebrae and has an incredible range of motion [2] with each two vertebrae capable of lateral flexion, rotations, flexion, and extension to each other. State-of-the-art methods unfortunately do not come close to capture these degrees of freedom. The aforementioned Xsens MVN Link suit consists of 17 inertial measurement units (IMUs), measuring local orientations from which the pose is derived. Considering the spine, only 4 IMUs contribute to its pose estimation: one on each shoulder, one on the chest, and one on the pelvis. As will be discussed in Section 2, this is not sufficient. To allow capturing the high degree of freedom of the spine, the number of measurement points needs to be increased.

Kinesiology tapes are widely used in all sports mainly for muscle recovery and injury prevention of athletes. Punched or perforated kinesiology tape, as can be seen in the background of Fig. 2, is used because of greater stretch capacity and breath-ability. In this work, we divert this kind of kinesiology tape from its intended use and instead apply it in a marker-based motion capture system. As can be seen in Fig. 1, the tape is fitted to the back of an athlete and serves as an easy-to-detect pattern with the dots of the tape used as markers in a multi-view camera system.

The key contributions of this paper are:

- The introduction of highly suitable marker for tracking in sports, that can be fast and easily attached and do not restrict movements
- An adopted motion capture framework, that uses structural a priori knowledge in 3D instead of image-based feature encoding and matching.
- A high precision tracker that robustly tracks dense markers and is capable of handling occlusions

2. Related Works

Accurately tracking the human spine in motion is still an open problem in computer vision. While today human motion capture is often solved marker-less using for instance a combination of key point networks, such as the Convolutional Pose Machines by Wei *et al.* [25] and a 2D-to-3D pose-lifting network as Bogo *et al.* [3], such a trend cannot be observed for human spine capturing. This is not surprising, since the back has few cues in its local environment that can be used to infer internal motion. To the best of our knowledge, there is currently no dedicated marker-less spine motion capture system. A major limitation of general marker-less motion capture systems is that applied 3D skeletons typically consist of 1 to 3 joints for the complete spine, which cannot accurately represent its various contortions.

However, there are a number of methods to infer the motion of the spine using additional technical tools. The low-cost device of Kam *et al.* [12] infers the total curvature of the spine by the outcoupling of light of an optical fiber attached to the back, but it cannot give a precise 3D path.

Similar to the aforementioned Xsens MVN suit by Schepers *et al.* [21], the specialized inertial tracking device by Hajibozorgi and Arjmand [10] captures the movement of the spine via a series of four IMUs. Voinea *et al.* [24] use 5 IMUs and a 7th-grade polynomial is fitted to approximate the curvature of the spine. Yet, IMU-based systems suffer from drifting in space, since a two times integration leads to a quadratic error propagation with time.

With extensive hardware input, Dynamic FAUST by Bogo *et al.* [4] is able to build a high-precision motion and shape capture system. It uses 22 stereo cameras and 34 speckle projectors to capture 3D body shapes. In addition, body painted features are captured by 22 RGB cameras. The high-frequency texture is mapped onto the body scan and an optical flow-based texture registration provides dense matches. The occurring per-vertex 3D displacements can be used to optimize the body shape and track individual vertices over time. This 4D dynamic scanner creates sequences of 3D body shapes with consistent vertex IDs. Its localization precision, which is closer than 1 mm, is due to the massive amount of data that provides texture matches of images shot from very similar viewpoints.

The retro-reflective marker-based optical motion capture (OMC) system by Vicon [1] is often seen as gold-standard. The study by Merriault *et al.* [16] reports a mean 3D position error of 1.5 mm while Rast *et al.* [19] reports a range of 0.1 to 5.3 mm, varying with the size of the captured volume. In the specific case of 3D trunk movement capturing with 10 markers along the spine and 12 further on other trunk landmarks, Rast *et al.* [19] compare sources of errors, as the instrumental error of Vicon, different landmark protocols for the markers and soft tissue artifacts.

However, the study suffers from limitations, as experiments of the range of motion task for flexion and extension of the spine could not be recorded since the visibility of the markers was insufficient. A likely cause is the minimal number of two cameras needed to track. With the suggested default number of 3 cameras [1] in a 360-degree studio, this threshold might not be reached in a lot of frames, so that trajectories are lost.

The most similar approach to ours is a retro-reflective mesh suit by Hiroaki *et al.* [22], in which stripes are arranged as a grid and intersections serve as markers. The stripes have a distance of about 30 mm from each other, resulting in a high marker density of 0.16 markers/cm². Similar to our argumentation, the authors suggest that a high marker density can only be achieved if connectivity information is present, which they extract directly from the image. Stripe lines are traced and a 2D mesh is extracted, which is fused in 3D with information from other views. This workflow is fundamentally different from ours, as we use no active light setup and enable a higher marker density by explicitly omitting to find 2D connectivity information or any other 2D feature encoding. Instead in 2D we only detect positions and solve the connectivity problem in 3D exploiting a priori known structural information.

3. Method

Our marker-based tracker uses perforated kinesiology tape with a width of 5 cm and dots having a diameter of roughly 1 cm (see Fig. 2). Attached to the back it does not limit the movements of an athlete nor will it fall off in any circumstance. It has a unique pattern of alternating rows with two or three dots next to each other. As can be seen in the teaser in Fig. 1, the large difference in brightness helps in cases of underexposed photos thereby allowing for short exposure times of the camera. By their nature, circles are stable 2D markers and a robust center point extraction is possible even in very noisy or (motion) blurred images. The kinesiology tape provides a high marker density, with 22 dots on a 20×5 cm stripe resulting in a marker density of 0.22 markers/cm².

Our hardware setup consists of 6 global shutter RGB cameras (FLIR ORX-10g-51S5-C) set to a resolution of 2448×1600 and a frame rate of 100 fps. The exposure time of the camera is set to 1 ms, which is rather short and we consequently observe very little motion blur when testing with fast stripe movements. As can be seen in Fig. 3 the cameras are circularly arranged covering 180 degrees of the athlete. The cameras are synchronized and calibrated to each other and to a Vicon infrared marker reference system (Vicon Motion Systems, Oxford, UK [1]). The Vicon system uses 8×T010 cameras and Nexus software in version 1.8.5.

An overview of our signal processing pipeline is



Figure 3. As motivated in [20], an athlete is performing a handstand in a studio setup with 6 cameras in an u-shape arrangement covering 180 degrees. The kinesiology tape attached to the back can be seen in 4 views. In addition a reference Vicon [1] camera setup tracks infrared markers attached to the rest of the body.

illustrated as a block diagram in Fig. 4. As a first step, the stripe and its dots are detected in each camera individually. A Mask R-CNN as proposed by He *et al.* [11] is used as a stripe detector. The network is pre-trained on the MS COCO [13] dataset and we refine the network on a dataset consisting of 1988 training samples and 631 validation samples. Each sample is a tuple of an RGB image acquired in our studio and the corresponding mask. A detected mask m (see Fig. 4 a)), is used as a region of interest (ROI) for a blob detector (Fig. 4 b)), detecting the dots in the stripe. For our purpose, the blob analysis of the MATLAB Computer Vision Toolbox [23] is sufficient to provide blob center points $b \in \mathcal{B}$.

The points b are mutually triangulated using blobs of another view and the resulting 3D point is considered as a 3D candidate point $p \in \mathcal{P}$, if its back projection error is below a threshold th_1 . The resulting point cloud \mathcal{P} consists of points that can be seen by two cameras. Each point p_i has a set of attributes: the identifiers a_i of cameras used for triangulation, the blobs b_i used for triangulation and a normal n_i derived as the negative normalized view directions of participating cameras. Next, two points $p_1, p_2 \in \mathcal{P}$ are merged into a new point $p_3 = (p_1 + p_2)/2$, if the Euclidean distance towards each other is below threshold th_2 and if the back projection error of the merged

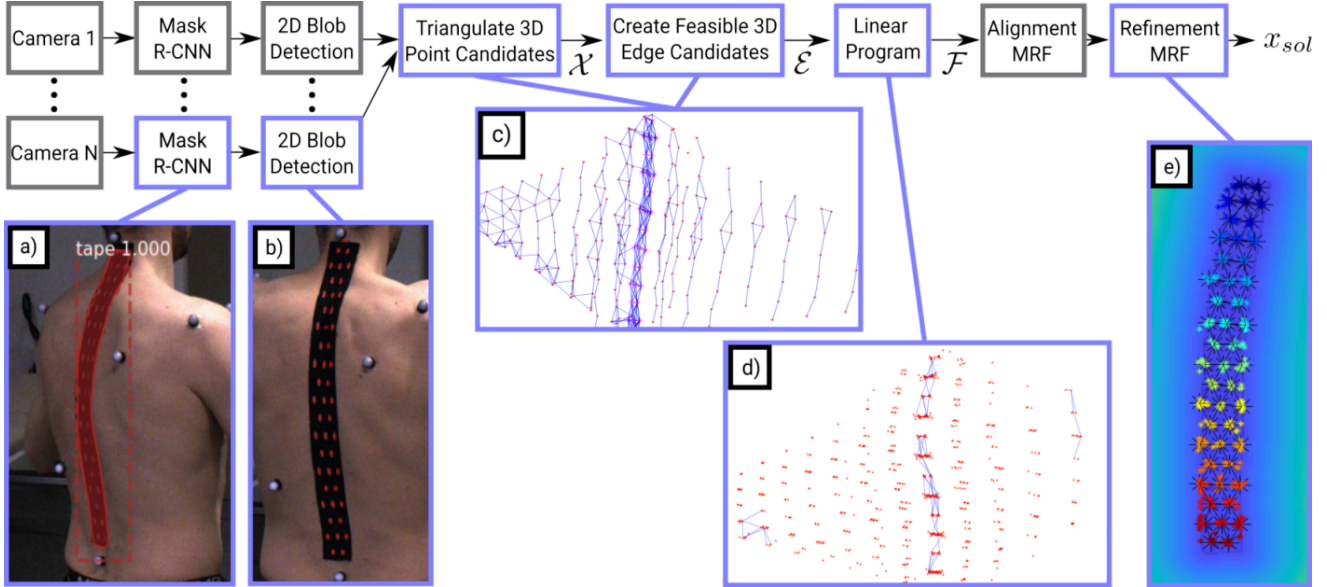


Figure 4. The proposed tracking pipeline: a) a Mask R-CNN [11] detects the kinesiology tape in N camera images. The corresponding mask is used as a region of interest (ROI) in which b) a blob detector finds the dots of the tape. c) dot marker points of all cameras are mutually triangulated which gives the 3D point candidate cloud \mathcal{X} , in which edges of appropriate lengths are found \mathcal{E} . d) a linear program selects most likely edges and gives a feasible edge set \mathcal{F} . A 3D-based alignment Markov Random Field (MRF) model is iteratively registered to \mathcal{F} given the a priori known structure of the dots inside of the tape. e) finally a second MRF optimized by particle belief propagation refines the model to all camera views and outputs the estimated 3D marker coordinates x_{sol} .

point in all views a_1 and a_2 is below the threshold th_1 . If p_1 and p_2 are merged, they are removed from \mathcal{P} and p_3 is added to \mathcal{P} with the attribute $a_3 = a_1 \cup a_2$ and b_3, n_3 respectively. This process is repeated as long as points can be merged. As the next step, the point cloud \mathcal{P} is transformed into an edge cloud \mathcal{E} with $\{e \in \mathcal{E} \mid l_e > th_3 \text{ and } l_e < th_4\}$, where l_e is the length of an edge connecting two points in \mathcal{P} . The thresholds th_3 and th_4 are the lower and upper boundaries of the a priori known distances of dots (see Fig. 2). In this way, we allow variations in distance between the dots of the tape if the skin at the back is stretched or contracted. The described sets can be seen in Fig. 4 c) with $p \in \mathcal{P}$ printed in Red and $e \in \mathcal{E}$ in Blue.

3.1. Linear Program

The task of the linear program is to select a subset from the set of edges \mathcal{E} as an approximation of the stripe (see Fig. 4 c) and d)). For this task, we define a graph $\mathcal{G} = (\mathcal{P}, \mathcal{E}, \mathcal{C})$ on the stripe, where \mathcal{C} is a set of costs associated to the points. As can be seen in Fig. 2, points $p \in \mathcal{V}$ are the dots of the stripe, and $e \in \mathcal{E}$ are edges connecting points. Given, that each stripe starts and ends with a two dots row, the number of target edges can be calculated by

$$n_e = \lfloor n_{row}/2 \rfloor \cdot 2 + \lceil n_{row}/2 \rceil + (n_{row} - 1) \cdot 6, \quad (1)$$

with $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ the floor and ceiling function. We assign individual edge costs $c_e \in \mathcal{C}$ as the inverse of the number of neighboring edges Ω_e connected to the current edge. In Fig. 2 the black edges of the graph are neighboring edges of the red edge e_{ij} , which is defined by the two points p_i and p_j . Consequently, Ω_e is defined as all edges connected to p_i or p_j except e_{ij} .

The linear program solves an edge selection task, with selection hypothesis \mathcal{H} and indicator variables x_e , which take value 1 if \mathcal{H} is selected, and 0 otherwise. The stripe detection task is to select hypotheses that minimize the total costs. This can be cast into the following binary optimization problem:

$$\arg \min_{x \in \mathcal{F}} \sum_{e \in \mathcal{E}} c_e x_e \quad (2)$$

where the feasibility set \mathcal{F} is subject to

$$\sum_{e \in \mathcal{E}} x_e \leq n_e, \quad (3)$$

$$\forall e \in \mathcal{E} : \sum_{\omega_e \in \Omega_e} \omega_e x_e \leq 12, \quad (4)$$

$$\forall e \in \mathcal{E} : \sum_{\omega_e \in \Omega_e} \omega_e x_e \geq 6, \quad (5)$$

The objective function in Eq. 2 minimizes the costs and thereby maximizes the number of neighboring edges. Eq. 3 limits the total number of selected edges to n_e . Eq. 4 and Eq. 5 constrain the number of neighboring edges to a range of 6 to 12, which as can be seen in Fig. 2 reflects the pattern of the dots.

In addition to Eq. 3 - 5, the following condition has to be satisfied: For all edge pairs $e_{ij}, e_{kl} \in \mathcal{E}$ that are non-identical ($i \neq k, j \neq l$) and are triangulated using the same blobs in one camera, i.e. $b_i = b_k, b_j = b_l$, one indicator x_{ij} or x_{kl} must be zero.

If the binary linear program is infeasible, which it may be in case of occlusions or missing blob detections, we iteratively decrease the number of target edges n_e until a feasible solution can be found. The resulting problem can be solved to optimality using BLP solvers like Gurobi [8].

3.2. Alignment Markov Random Fields (MRF1)

In the following, two Markov random fields MRF1 and MRF2 are sequentially applied (see Fig. 4). In MRF1 a model is optimized to match the edges of the feasibility set \mathcal{F} and in MRF2 the model is further refined to match the camera images.

The first Markov random field is a set of random variables called nodes $\mathbf{x} = (x, y, z, d) \in \mathcal{V}$, with x, y, z the 3D location of the node and a local distance d . Each node has a neighborhood \mathcal{N} , defining connections to other nodes, which consists of the solid and dashed edges drawn in Fig. 2. Since we expect a different amount of stretching of the stripe in longitudinal and transverse direction, we distinguish between longitudinal $\mathcal{N}_{1,s}$ and transverse $\mathcal{N}_{2,s}$ edges. Transverse edges connect points within a row, while longitudinal edges connect different rows. The random variables \mathbf{x} are optimized by minimizing the energy of the MRF model:

$$E(\mathbf{x}) = \sum_{s \in \mathcal{V}} \psi_s(\mathbf{x}_s) + \sum_{s \in \mathcal{V}} \sum_{t \in \mathcal{N}_s} \psi_{s,t}(\mathbf{x}_s, \mathbf{x}_t), \quad (6)$$

where $\psi_s(\mathbf{x}_s)$ is the unary potential and $\psi_{s,t}(\mathbf{x}_s, \mathbf{x}_t)$ the binary potential. The unary potential is defined as the shortest distance to a point \mathbf{p} of the feasibility set \mathcal{F} :

$$\psi_s(\mathbf{x}_s) = \min(\mathbf{x}_s, \mathbf{p}), \forall \mathbf{p} \in \mathcal{F} \quad (7)$$

The minimal energy in Eq. 6 is consequently close to the linear programs solution. The binary potential is defined as

$$\begin{aligned} \psi_{s,t}(\mathbf{x}_s, \mathbf{x}_t) = & \Theta_1 (\|\mathbf{p}_{st}\|_2 - d)^2 + \Theta_2 \|\langle \mathbf{p}_{st}, \mathbf{n}_{s,t} \rangle\|_2^2 \\ & + \Theta_3 \exp(d_{min} - \|\mathbf{p}_{st}\|_2)^2 \mathbb{I}[\|\mathbf{p}_{st}\|_2 < d_{min}] \\ & + \Theta_4 (\|\mathbf{p}_{st}\|_2 - d_{target})^2 \end{aligned} \quad (8)$$

where $\mathbf{p}_{st} = \mathbf{p}_s - \mathbf{p}_t$ and $\mathbb{I}[\cdot]$ is the Iverson bracket, which is 1 if its argument is true and zero otherwise. The

parameters $\Theta_1 - \Theta_4$ weigh individual terms. In the first term, Θ_1 balances the random variable d to match the node distance between \mathbf{p}_s and \mathbf{p}_t . The second term pushes the orientation of the connection \mathbf{p}_s to \mathbf{p}_t to be orthogonal to nearby normal $\mathbf{n}_{s,t} \in \mathcal{N}_f$, where the set \mathcal{N}_f is a smooth field of normals derived from the point attributes \mathbf{n}_i . In \mathcal{N}_f each camera is equally weighted, yet, individual weights for each camera would be beneficial, as could be derived from an advanced clustering of the corresponding cameras as in Dockhorn *et al.* [7]. The third term encourages \mathbf{x}_s and \mathbf{x}_t to keep a minimal distance d_{min} from each other and the fourth term constrains the distance \mathbf{p}_{st} to match the target length d_{target} , which are measured tape distances that can be estimated from Fig. 2.

The MRF is an iterative algorithm that needs an initialization of a template, which is a complete structure of the stripe including the target number of points with corresponding edges. In order to position the structure in 3D we fit a third degree polynomial to the feasibility set \mathcal{F} . Together with the normal field \mathcal{N}_f we can create a tape template along the run of the polynomial. The long connections in Fig. 2 that span across two rows or columns are included to avoid a folding of the structure of the stripe.

In order to efficiently minimize the MRF energy (Eq. 6), we approximate the maximum a posteriori (MAP) probability using max-product particle belief propagation as Pacheco *et al.* [18] and Hachmann *et al.* [9]. Particles are estimated using slice sampling as in Müller *et al.* [17]. In order to improve convergence, the slice sampling particle set $P_t = \{\mathbf{x}_t^{(1)}, \dots, \mathbf{x}_t^{(p)}\}$ of node t is augmented:

$$P_{t,aug} = P_t \cup P_{mobility} \cup P_{knn}. \quad (9)$$

The set $P_{mobility}$ adds the position of neighboring nodes which increase stripe movements and thus convergence and P_{knn} adds the k-nearest points from the feasibility set \mathcal{F} as these are likely candidates for a good position.

3.3. Refinement Markov Random Field (MRF2)

The second MRF uses the same energy formulation as Eq. 6 and is initialized by the resulting positions of MRF1. MRF2 is computationally much more demanding, as it tends to need more iterations to converge. That is why we decrease the search space and reduce the random variable \mathbf{x} to the 3D positions of the nodes. For the same reason, the number of edges is reduced to only include the solid connections in Fig. 2. The unary potential of MRF2 is image-based:

$$\psi_s(\mathbf{x}_s) = \sum_{c \in \mathcal{C}} I_c(\Phi_c(\mathbf{x}_s)) \quad (10)$$

Here, $\Phi_c(\cdot)$ is the back projection of 3D points into image coordinates of camera $c \in \mathcal{C}$. The 2D image I_c , which can

be seen in the background of Fig. 4 e), contains the inverse image acquired by camera c inside of the Mask R-CNN mask m and is filled with increasing values on the outside of m , which are calculated by a distance transformation. The binary potential of MRF2 is similar to Eq. 8:

$$\begin{aligned} \psi_{s,t}(\mathbf{x}_s, \mathbf{x}_t) = & \Theta_5(I_c(\Phi_c(\mathbf{x}_s)) + I_c(\Phi_c(\mathbf{x}_t)) - 2I_c(\Phi_c(\mathbf{x}_{st}))) \\ & + \Theta_6 \|\langle \mathbf{p}_{st}, \mathbf{n}_{s,t} \rangle\|_2^2 + \Theta_7 \|\langle \mathbf{p}_{st}, \alpha_{s,t} \rangle\|_2^2 \\ & + \Theta_8 (\|\mathbf{p}_{st}\|_2 - d_{target})^2 \end{aligned} \quad (11)$$

with \mathbf{x}_{st} the arithmetic mean of \mathbf{x}_s and \mathbf{x}_t . Different to Eq. 8 the first term calculates a relation between the nodes positions and an intermediate point \mathbf{x}_{st} : Given that the appearance of the stripe dots are bright and regions in between are dark. Furthermore, the third term aligns the model to the currently estimated direction $\alpha_{s,t}$, that is either longitudinal or transversal. The corresponding dot product between the edge \mathbf{p}_{st} and the identified direction $\alpha_{s,t}$ is penalized. The directions in $\alpha_{s,t}$ are derived from the normal field \mathcal{N}_f and the fitted polynomial. The same particle set $P_{t,aug}$ as in Eq. 9 is used and the back projection $\Phi_c(P_{t,aug})$ it is illustrated in the foreground of Fig. 4 e) as colorful markers. The parameters as $\Theta_1, \dots, \Theta_8$ and d_{min} are estimated empirically.

After convergence of MRF2 the final positions of the nodes \mathbf{x}_{sol} can be fitted to a musculoskeletal model like OpenSIM [6] for bio-mechanical analysis. We share software examples of our tracking framework on GitHub¹.

4. Experiments and Results

We conduct a series of experiments on artificial and studio sequences to prove the effectiveness of our method. While the overall framework remains unchanged, each domain has different empirically estimated parameters. The *Artificial* dataset is the 548 frames long sequence *50009_hips* of the Dynamic FAUST dataset by Bogo *et al.* [4], which shows the largest amount of spine movements among them. The coordinate system suggests that the height of the body shape is 1.75, which we set to 1.75 meters in order to calculate reasonable errors in millimeters.

Experiment 1: Using the artificial sequence we compare our method to an arbitrary keypoint-based approach that is able to robustly track 4 to 8 keypoints positioned left and right next to the spine. These points are used to fit 1 to 3 planes that approximate the back, which can be seen in Fig. 5. The shortest distance of the planes to the tape dot ground truth is measured, which is a one-sided Hausdorff distance. The errors of planes and our approach can be seen in the time series in Fig. 6 and mean values can be found in the top row of Table 1.

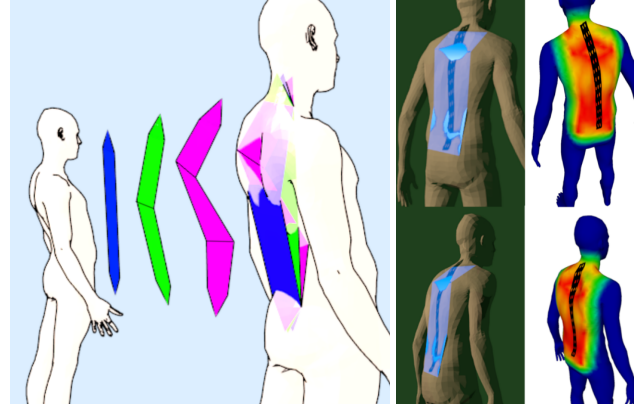


Figure 5. (left) We perform experiments with one (blue) to three (magenta) planes fitted to markers left and right next to the spine. (center) Qualitative renderings for a 3-plane approximation with the clearly visible plane-to-body shape intersections. (right) Hausdorff distance color encoding with red values being a close approximation to green values indicating a large distance to the planes. This sparse marker-based spine approximation is evaluated on the artificial test sequence and the closest distance of the plane to the 3D ground truth is plotted in the time series in Fig. 6.

Experiment 2: We added skin-colored balls falling from the sky to simulate arbitrary occlusions. As illustrated on top of Fig. 7, in a rendered scene the red and green marked ground truth positions and occlusions can be easily calculated. The occlusions are mapped to an artificial stripe pattern in the middle showing the number of views that can see a marker. Occlusions typically occur as burst

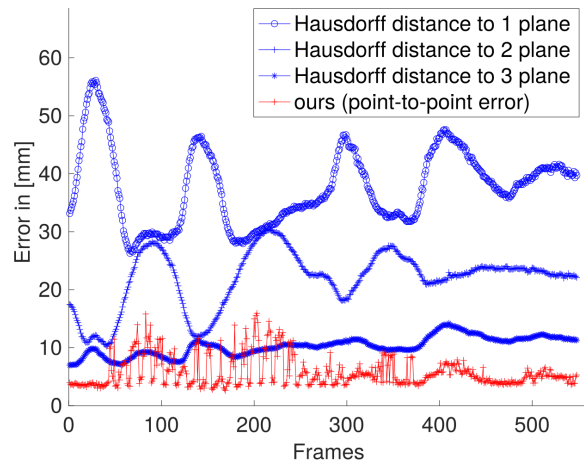


Figure 6. Comparison of the 1 to 3 planes approximation to the ground truth with our marker point to ground truth point error on the 0 balls artificial sequence. Even though a point to point error is more restrictive than a plane to point error, our method shows that it is more accurate.

¹<https://github.com/hendrik-hachmann/spinemocap>

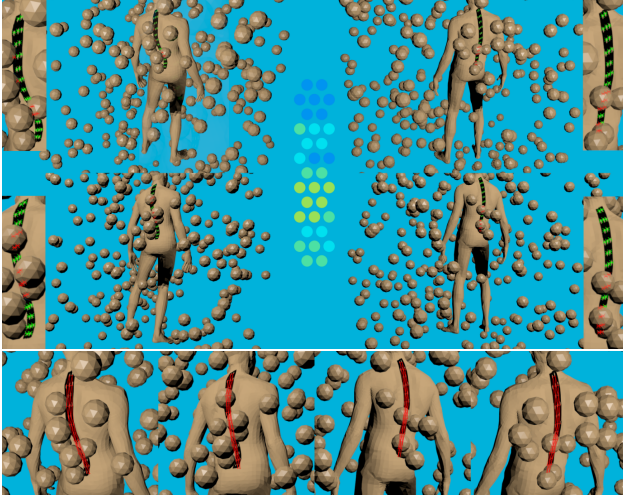


Figure 7. Occlusion experiment: (top rows) 4 views of the *Artificial* sequence with 5000 particle system balls falling from the sky and creating occlusions. Green or red markers on the stripe symbolize whether the particular dot can be seen by the particular view. The schematic in the middle shows the number of cameras that can see the markers with dark Blue meaning that all cameras can see the dot and yellow meaning that only one camera can see the dot. (bottom row) Corresponding spine tracker result. It can be observed that the tracker can estimate the position of stripe markers quite well even though this is a case with a heavy amount of occlusions. While in this case, occlusions are caused by balls, self-occlusions are very common, as in a 180 degrees studio setup the total stripe can never be seen by all cameras at once.

Dataset	1-plane	2-planes	3-planes	MoSh [14]	ours
Artificial ↓	37.67	21.89	10.29	-	5.55
Studio ↓	-	-	-	24.75	4.12

Table 1. Quantitative tracking results: The dataset *Artificial* is a 4D sequence of the Dynamic FAUST dataset [4] with kinesiology tape texture (see Fig. 7). The 1-3 plane(s) results are Hausdorff distance errors of the planes illustrated in Fig. 5 to the ground truth markers. Error-values in [mm] are calculated with the virtual body size set to 1.75 meters. The *Studio* dataset is a studio sequence which can be seen in Fig. 11. MoSh [14] denotes a motion and shape fitting to a sparse Vicon [1] marker trajectory. Individual marker positions are inferred from the texture fitting as illustrated in Fig. 8.

occlusions, meaning that several dots close to each other are occluded at the same time. A qualitative result can be seen at the bottom of Fig. 7. Even with the high number of occlusions in this frame, the marker position estimation stays robust. Quantitative results can be found in Table 2.

Experiment 3: We captured a studio sequence with major spine movement, which we call *Studio* using our RGB cameras and Vicon. For every 50th frame, we create

Artificial dataset	0 balls	5000 balls	10000 balls
4 cameras	17536	11197	6846
3 cameras	0	5328	6812
2 cameras	0	825	2764
1 camera	0	83	607
0 cameras	0	103	507
error [mm] ↓	5.55	6.17	7.89

Table 2. Occlusion experiment results corresponding to Fig. 7: (top) Visibility of markers in the *Artificial* sequence with 0, 5000, and 10000 balls as occluders. 4 cameras mean no occlusions, as 4 cameras can see the markers. 0 cameras mean that a marker is not visible in any view. (bottom) Marker point-to-ground truth point tracking the error in [mm].

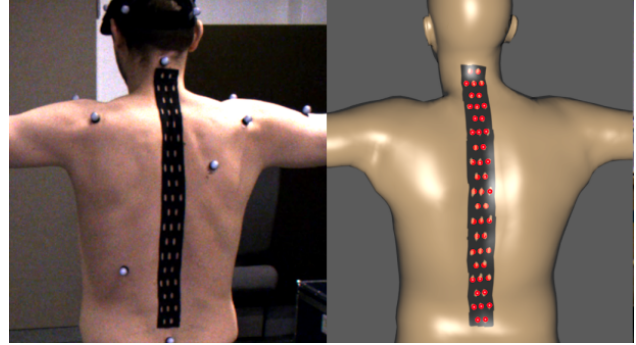


Figure 8. Vicon marker-based SMPL fitting with stripe texture mapping. (left) T-pose reference frame (right) SMPL fit with texture and red balls as 3D ground truth.

a 3D ground truth by triangulation of manually annotated images of the six cameras. Then we optimize a body shape to the sparse Vicon marker set similar to MoSh by Loper *et al.* [14]. As can be seen in Fig. 8, on a t-pose frame an artificial stripe texture is registered to the ground truth rendered as red balls. As SMPL [15] body shapes have consistent vertex IDs along the sequence, from the texture we can calculate a 3D estimate of the tape dots. Ground truth marker-to-marker estimates of this method and our tracker can be found in Table 1.

Qualitative results: In Fig. 7, 9, 10 and 11 some samples of our tracker on various studio sequences are visualized. Fig. 10 shows an artificial occlusion by a jogging bra, which occludes dots in all views. Even though the Mask R-CNN results are incomplete or include a second faulty detection, the tracking result is precise.

Discussion: The proposed tracking pipeline in Fig. 4 is different from many other trackers, as it has no image-based feature encoding and matching, such as the optical flow texture matching in Dynamic FAUST [4] or the image-based connectivity extraction in the mesh suit approach by Hiroaki *et al.* [22]. We claim that since we have only a detector, we can increase the marker density to a higher

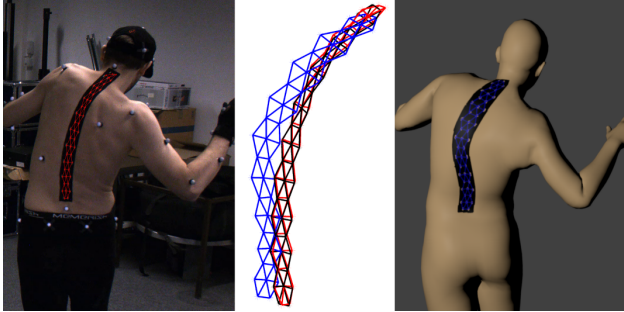


Figure 9. (left) Studio sequence image with the result of our method (center) Comparison of results of MoSh printed in Blue, our method in Red to the ground truth in Black (right) Vicon marker-based SMPL fitting with tracking result in Blue.



Figure 10. The proposed spine tracker also works with jogging bras. Even though some dots in the stripe are occluded and the Mask R-CNN detections have a second faulty detection (right side), the overall stripe fitting is accurate.

Properties	4D [4]	IMU [24]	Vicon [1]	OF [12]	MS [22]	ours
model preparations	(--)	(o)	(-)	(o)	(++)	(++)
precision	(++)	(o)	(+)	(--)	(+)	(+)
comfort	(++)	(-)	(-)	(-)	(++)	(++)
occlusion handling	(++)	(++)	(--)	(++)	(--)	(+)
outdoor	(--)	(++)	(--)	(++)	(-)	(+)

Table 3. Qualitative comparison of various motion capture systems (estimated by the authors): 4D denotes body painting in Dynamic FAUST [4], IMU denotes inertial measurement units [24], OF denotes optical fibers [12] and MS denotes a mesh suit [22].

level. In comparison to Hiroaki *et al.* [22], who report a marker density of 0.16 marker/cm², using a similar studio setup we achieve a density of 0.22 marker/cm².

We would like to conclude this section with a comparison of different spine motion capture systems. In Table 3 strengths and weaknesses regarding certain properties are listed. While some comparisons are difficult to assess, the following can be probably agreed upon: Dynamic FAUST is rather precise, and active lightning systems may run into problems if used outdoors. Focussing on our approach we claim that a model preparation of a few seconds is rather short. The precision is in the upper mid-range. The term comfort refers to whether natural

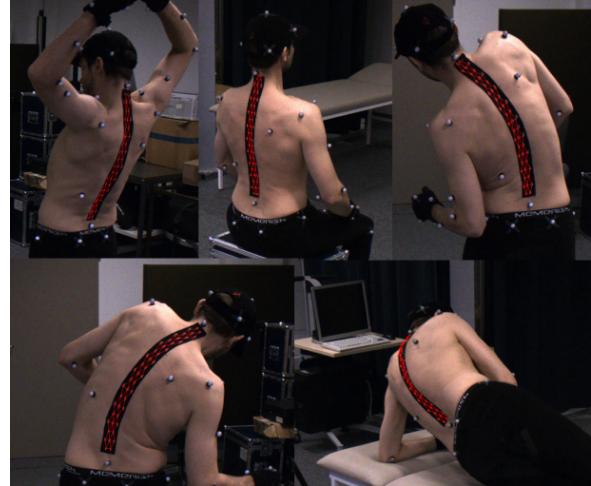


Figure 11. Tracker results on studio samples.

movements are possible and includes if a device stands out, falls off, or is heavy. We showed that our framework is capable of occlusion handling with interpolated marker estimation even if multiple markers are occluded in all cameras. Finally, given that cameras come with automatic exposure, the combination with a high contrast marker allows for outdoor use. We want to stress that our tracker uses intra-frame prediction only and no initial pose is given.

5. Conclusion

We introduced a kinesiology tape-based tracking system that is able to capture the human spine in 3D over time. The tape is highly flexible and can follow any spine movements. Several experiments on artificial and studio sequences show that our tracking framework can reliably track markers with high density and is robust in case of occlusions. The method is not limited to the application of spine motion capture and can be adopted to all other sorts of surface tracking. We assume that the tracking performance can be further improved by temporal context, which we intend to investigate in future work.

Acknowledgements

This work was supported by the Federal Ministry of Education and Research (BMBF), Germany under the project LeibnizKILabor (grant no. 01DD20003) and the AI service center KISSKI (grant no. 01IS22093C), the Center for Digital Innovations (ZDIN) and the Deutsche Forschungsgemeinschaft (DFG) under Germany's Excellence Strategy within the Cluster of Excellence PhoenixD (EXC 2122). The authors also thank Dr. Oliver Müller for fruitful discussions and hints.

References

- [1] Vicon Tracker User Guide. <https://docs.vicon.com/>, 9 2020. 2, 3, 7, 8
- [2] Nikolai Bogduk. *Clinical anatomy of the lumbar spine and sacrum*. Elsevier Health Sciences, 2005. 2
- [3] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J. Black. Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In *Computer Vision – ECCV 2016*, Lecture Notes in Computer Science, pages 561–578. Springer International Publishing, Oct. 2016. 2
- [4] Federica Bogo, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Dynamic FAUST: Registering Human Bodies in Motion. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5573–5582, 2017. 2, 6, 7, 8
- [5] Amity Campbell, Peter O’Sullivan, Leon Straker, Bruce Elliott, and Machar Reid. Back pain in tennis players: a link with lumbar serve kinematics and range of motion. *Medicine and science in sports and exercise*, 46(2):351–357, February 2014. 1
- [6] Scott L. Delp, Frank C. Anderson, Allison S. Arnold, Peter Loan, Ayman Habib, Chand T. John, Eran Guendelman, and Darryl G. Thelen. OpenSim: Open-Source Software to Create and Analyze Dynamic Simulations of Movement. *IEEE Trans. Biomed. Eng.*, 54(11):1940–1950, 2007. 1, 6
- [7] Alexander Dockhorn, Christian Braune, and Rudolf Kruse. Variable density based clustering. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8, Dec. 2016. 5
- [8] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2023. 5
- [9] Hendrik Hachmann, Benjamin Krüger, Bodo Rosenhahn, and Waldo Nogueira. Localization of cochlear implant electrodes from cone beam computed tomography using particle belief propagation. In *International Symposium on Biomedical Imaging, ISBI*, april 2021. 5
- [10] M. Hajibozorgi and N. Arjmand. Sagittal range of motion of the thoracic spine using inertial tracking device and effect of measurement errors on model predictions. *Journal of Biomechanics*, 49(6):913–918, 2016. SI: Spine Loading and Deformation. 2
- [11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. *CoRR*, abs/1703.06870, 2017. 3, 4
- [12] Wern Kam, Kieran O’Sullivan, Mary O’Keeffe, Sinead O’Keeffe, Waleed S. Mohammed, and Elfed Lewis. Low cost portable 3-D printed optical fiber sensor for real-time monitoring of lower back bending. *Sensors and Actuators A-physical*, 265:193–201, 2017. 2, 8
- [13] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft COCO: Common Objects in Context, 2014. 3
- [14] Matthew Loper, Naureen Mahmood, and Michael J. Black. MoSh: Motion and Shape Capture from Sparse Markers. *ACM Trans. Graph.*, 33(6), nov 2014. 7
- [15] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A Skinned Multi-Person Linear Model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, Oct. 2015. 7
- [16] Pierre Merriault, Yohan Dupuis, Rémi Boutteau, Pascal Vasseur, and Xavier Savatier. A Study of Vicon System Positioning Performance. *Sensors*, 17(7), 2017. 2
- [17] Oliver Müller, Michael Y. Yang, and Bodo Rosenhahn. Slice Sampling Particle Belief Propagation. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1129–1136, Dec. 2013. 5
- [18] Jason Pacheco and Erik Sudderth. Proteins, Particles, and Pseudo-Max-Marginals: A Submodular Approach. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2200–2208, Lille, France, 07–09 Jul 2015. PMLR. 5
- [19] Fabian M. Rast, Eveline S. Graf, André Meichtry, Jan Kool, and Christoph M. Bauer. Between-day reliability of three-dimensional motion analysis of the trunk: A comparison of marker based protocols. *Journal of Biomechanics*, 49(5):807–811, 2016. 2
- [20] B. Rosenhahn, C. Schmaltz, T. Brox, J. Weickert, D. Cremers, and H.-P. Seidel. Markerless motion capture of man-machine interaction. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 3
- [21] Martin Schepers, Matteo Giuberti, and Giovanni Bellusci. Xsens MVN: Consistent Tracking of Human Motion Using Inertial Sensing. *White paper*, 03 2018. 1, 2
- [22] Hiroaki Tanie, Katsu Yamane, and Yoshihiko Nakamura. High marker density motion capture by retroreflective mesh suit. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation, ICRA 2005, April 18-22, 2005, Barcelona, Spain*, pages 2884–2889. IEEE, 2005. 3, 7, 8
- [23] Inc. The MathWorks. *Computer Vision Toolbox*. Natick, Massachusetts, United State, 2019. 3
- [24] Gheorghe-Daniel Voinea, Silviu Butnariu, and Gheorghe Mogan. Measurement and Geometric Modelling of Human Spine Posture for Medical Rehabilitation Purposes Using a Wearable Monitoring System Based on Inertial Sensors. *Sensors*, 17(1), 2017. 2, 8
- [25] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional Pose Machines. *CoRR*, abs/1602.00134, 2016. 2