Machine Vision and Applications
DOI 10.1007/s00138-011-0371-7

ORIGINAL PAPER

3-D–2-D spatiotemporal registration for sports motion analysis

Wee Kheng Leow · Ruixuan Wang · Hon Wai Leong

Received: 26 November 2010 / Revised: 9 July 2011 / Accepted: 31 August 2011

Abstract Computer systems are increasingly being used for sports training. Existing sports training systems either require expensive 3-D motion capture systems or do not provide intelligent analysis of user's sports motion. This paper presents a framework for affordable and intelligent sports training systems for general users. The user is assumed to perform the same type of sport motion as an expert, and therefore the performer's motion is more or less similar to the expert's reference motion. The performer's motion is recorded by a single stationary camera, and the expert's 3-D reference motion is captured only once by a commercial motion capture system. Under such assumptions, sports motion analysis is formulated as a 3-D–2-D spatiotemporal motion registration problem. A novel algorithm is developed to perform spatiotemporal registration of the expert's 3-D reference motion and a performer's 2-D input video, thereby computing the deviation of the performer's motion from the expert's motion. The algorithm can effectively handle ambiguous situations in a single video such as depth ambiguity of body parts and partial occlusion. Test results on Taichi and golf swing motion show that, despite using only single video, the algorithm can compute 3-D posture errors that reflect the performer's actual motion error.

Keywords Human motion analysis · Spatiotemporal registration · Dynamic programming · Belief propagation

W. K. Leow · R. Wang (✉) · H. W. Leong
Department of Computer Science, National University of Singapore, Computing 1, Singapore 117590, Singapore
e-mail: ruixuanwang@hotmail.com

W. K. Leow
e-mail: leowwk@comp.nus.edu.sg

H. W. Leong
e-mail: leonghw@comp.nus.edu.sg

1 Introduction

Computer systems are increasingly being used for sports training. Two kinds of computer-aided sports training systems are commercially available: 3-D motion-based systems and 2-D video-based systems. A 3-D motion-based system [1,2] uses multiple cameras to track the motion of reflective markers attached to the performer's body. The markers' 3-D positions are recovered and used to compute the performer's 3-D motion, which can be analyzed by the coach or compared with a 3-D reference motion of an expert. Such a system can provide an accurate motion analysis. However, it is very expensive and difficult to use for the general users.

A 2-D video-based system [3–6] captures the performer's motion using an off-the-shelf video camera and loads the video into a computer system. The system displays the performer's video and a pre-recorded expert's video side by side, and provides tools for the user to manually compare the performer's motion with the expert's motion. The system is affordable to general users. However, it cannot perform detailed motion analysis automatically.

To overcome the shortcomings of existing systems, this paper proposes a framework for affordable and intelligent sports training systems for general users that require only single stationary camera to record the user's motion. Sports motion analysis is formulated as a 3-D–2-D spatiotemporal motion registration problem (Sect. 3). A novel algorithm is developed to perform spatiotemporal matching of the 3-D reference motion of an expert and the 2-D input video of a performer, thereby computing the deviation of the performer's motion from the expert's motion (Sects. 4–7). The algorithm can effectively handle ambiguous situations in single video such as depth ambiguity of body parts and partial occlusion. It can be applied to analyze different types of sports motion. Test results on Taichi and golf swing motion show that the algorithm can compute 3-D posture errors that reflect the performer's actual motion error using only single video. The proposed framework can be potentially extended to become more feasible for real practical applications (Sect. 10).

2 Related work

Our 3-D–2-D spatiotemporal registration problem for sports motion analysis is closely related to several known research topics, namely human body tracking, human posture estimation, and video sequence alignment. However, there are fundamental differences between them. Human body tracking [7–9], in general, performs spatial matching between consecutive images in the input sequence without using 3-D reference motion. Human posture estimation infers the 2-D or 3-D body posture from a single or multiple images without solving temporal correspondence. Human body tracking methods often apply human posture estimation techniques [10–12]. Video sequence alignment [13,14] solves for the temporal correspondence between two sequences without posture matching and 3-D motion information. Our proposed problem involves both temporal correspondence and posture matching, which is much more complex than the related problems. In the following, existing work in the most related area, human posture estimation, is discussed more in detail.

The approach to human posture estimation can be roughly divided into two categories according to whether a 3-D human body model is pre-defined or not [15].

2.1 Approach without 3-D human body model

This approach does not use explicit human body model. It includes three main kinds of methods: mapping function-based methods, exemplar-based methods, and probabilistic assemblies of parts.

Mapping function-based methods [16–25] learn a nonlinear mapping function to map from image features to body postures. Then, the trained mapping function can directly determine the body posture from a single image. For example, Agarwal and Triggs [16] used 100-dimensional input vector that encodes local shapes of a human image silhouette, and 55-dimensional vector to represent 3-D full-body posture. Given a set of labelled training examples, they used relevance vector machine (RVM) [26] to learn a nonlinear mapping function that consists of a set of weighted basis functions. RVM has been extended to multivariate RVM for posture estimation [24]. More recently, Bissacco et al. [17] used a set of oriented Haar features to extract low-level motion and appearance information from images, and developed a multi-dimensional boosting regression technique to learn the mapping from Haar features to 3-D body postures. Urtasun and Darrell [25] used online local Gaussian processes (GP) to efficiently learn a multi-modal mapping from silhouette features to 3-D body postures. Ning et al. [20] and Sminchisescu et al. [23] used Bayesian mixture of experts (BME) to learn the mapping. Fossati et al. [19] used Gaussian process to learn the mapping from 2-D or 3-D parameterized trajectories of feet or hands to 3-D posture sequences for certain types of motions like skating and golfing.

Another example is manifold-based mapping [18,27,28]. A manifold is a topological space that is locally Euclidean. If the input image comes from a known type of 3-D motion model (e.g., walking), the 3-D motion model can be represented as a nonlinear manifold in a high-dimensional space. By mapping the manifold into a lower-dimensional space using embedding technique, and learning two nonlinear mappings between the embedded manifold and the visual input (i.e., silhouette) space and 3-D body posture space, 3-D body posture can be estimated from each input image by the two mapping functions.

Instead of training a mapping function, exemplar-based methods store a set of exemplar images with known 3-D postures, and estimate the posture in the input image by searching for the exemplar that is most similar to the input image [29–36]. Since multiple postures may have very similar images, the methods often output multiple 3-D posture estimations for the input image [34]. Since matching the image and each exemplar is often computationally expensive, researchers often save the computation time by constructing an embedding [29,32,33]. The embedding techniques [37,38] map a point in the image space into another low-dimensional space such that the similarity measurement between images can be efficiently computed in the embedded space. For example, Athitsos et al. [29] used AdaBoost to construct an embedding, by combining a set of 1-D embeddings that preserve rankings of the similarities between any input image and all exemplars in the embedded space.

Both mapping function-based and exemplar-based methods are useful only for a small set of body postures due to the complexity of human postures. They can recover only the body postures that are similar to those in the training images and exemplars.

Instead of matching the exemplar images globally, the methods of probabilistic assemblies of parts apply low-level feature detectors to detect likely body parts, and assemble them to obtain the 2-D body posture that best matches the detected features [39–49]. Individual body parts are detected using 2-D shape [48], SVM classifiers [49], AdaBoost [39,43], locally initialized appearance models [46], and motion of Kanade-Lucas-Tomasi (KLT) features [50]. For example, Mikolajczyk et al. [44] introduced a robust AdaBoost part detector to provide coarse 2-D localizations of body parts in the image. Once body part candidates are detected, body postures are assembled from the part candidates by applying prior knowledge or constraints such as joint connectivity and length ratio between parts. Mori [45] used superpixels as the element to represent the input image. Based on the boundaries of superpixels and constraints between the body parts, a rough 2-D posture configuration was obtained. Ren et al. [47] used pairwise constraints between body parts to assemble detected body parts into 2-D pose configurations. These pairwise constraints include aspect ratio, scale, appearance, orientation, and connectivity. Ramanan et al. [46] learned a global body part configuration model based on conditional random fields to simultaneously detect all body parts. Ferrari et al. [40] detected possible part positions by 'image parsing' [51] and disambiguate posture estimates by assuming that the appearance and position of body parts changes smoothly between subsequent frames. Andriluka et al. [39] learned a probabilistic appearance model for each body part based on Adaboost classifier's output, and combined such appearance models with the kinematic prior on the whole body configuration for people detection and posture estimation. Yao and Li [52] used context information to obtain more accurate estimates especially when self-occlusion appears in images. These methods can potentially estimate 2-D body postures in cluttered scenes, but they can only estimate postures approximately due to cluttered background and lack of constraints from structure.

2.2 Approach with 3-D human body model

This approach estimates body posture by synthesizing possible postures from a 3-D human model and matching them to the input images. It can be divided into two main classes of methods: continuous methods and probabilistic methods.

Continuous methods [7,53–56,11] minimize the error between the synthesized image and the real input image and applies continuous optimization algorithms to determine the locally optimal solution. Many continuous optimization algorithms can be used. For example, Bregler and Malik [7] used Quasi-Newton method for 3-D posture estimation. Ju et al. [55] used a gradient descent method for 2-D posture estimation. Rehg and Kanade [56] used Levenburg–Marquardt to estimate 3-D articulated posture. Continuous methods cannot guarantee that the solutions are globally optimal.

Probabilistic methods, which include particle filtering (CONDENSATION), Markov Chain Monte Carlo [57], and Belief Propagation [58–60,46,61,62], use sampling techniques to estimate postures. With enough samples, these methods can potentially obtain the globally optimal solution. The main difficulty of these methods is to search a very high-dimensional space for the globally optimal solution. To tackle this problem, belief propagation (BP) decomposes the high-dimensional search problem into a set of interrelated low-dimensional problems by iteratively estimating the pose distribution of each body part and propagating messages to its neighboring body parts. Hua et al. [59] applied BP to estimate 2-D body posture without self-occlusion. Sudderth et al. [62] used it for 3-D-articulated hand tracking from a single video. BP method is adapted and extended in our algorithm framework.

In order to obtain globally optimal solution, a strong prior model is often used to reduce the search space for posture estimation [63–66,10,8,67–70]. The prior model is often learnt from 3-D motion data captured by a commercial system. For example, Rutasun et al. [70] and Gupta et al. [65] used scaled Gaussian process latent variable models (SGPLVM) to learn a prior distribution of postures in a low-dimensional embedding space. Li et al. [10] used locally linear coordination (LLC) to learn a set of prior posture clusters in an embedding space. Fossati and Fua [64] obtained more accurate and realistic results by forcing the facing orientation of human body to be consistent with its motion direction. In addition, Brubaker et al. [63] used physics-based approach to model the dynamics of lower body parts in walking or running motion, and Taylor et al. [71] demonstrated that a binary latent variable model called implicit mixture of conditional restricted Boltzmann machines (imCRBM) worked effectively as a motion prior for 3-D human tracking.

3 Problem formulation

To clearly describe the problem, it is necessary to first describe the inputs of the problem, which consist of 3-D reference motion of the expert and 2-D input video of the performer (Sects. 3.1 and 3.2), and the complex relationships (Sect. 3.3) between them.

3.1 3-D reference motion

The 3-D reference motion of the expert includes:

1. Time-independent component: human body model.
   The human body model consists of a hierarchical skeleton model of bones and joints, and a triangular mesh model for the shapes of the body parts (Fig. 1b). The mesh model is divided into a set of mesh parts and each mesh part corresponds to one unique body part (Fig. 1b, c).
2. Time-dependent component: 3-D motion data.
   The 3-D motion data comprise a temporal sequence of global positions of human body in the world coordinate system, and joint angles of the body parts. These data define the *reference posture* (Fig. 4c) at time t denoted as B_t. For any reference posture B_t, the joint angles will be used to articulate the skeleton model, and the mesh part associated with each body part will be rotated accordingly. The sequence of B_t, $t \in \mathcal{T} = \{0, \ldots, L\}$, together with the human body model, defines the 3-D reference motion.

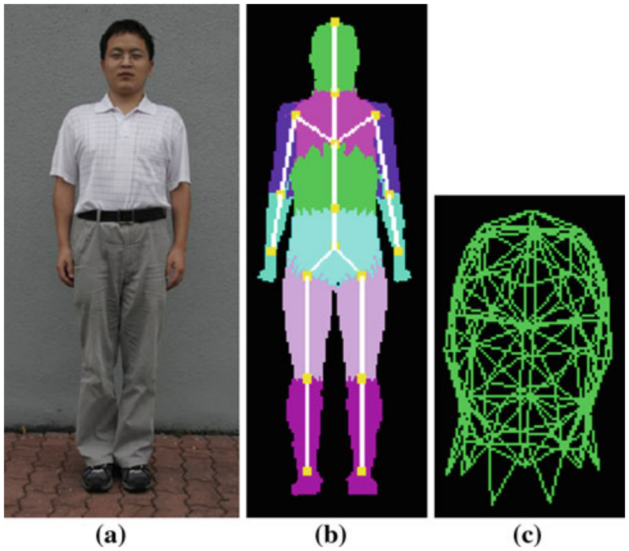


Fig. 1 Human body model. **a** A performer's human body image. **b** Human mesh model adapted to fit shape and size of the performer's human body, and human skeleton model including joints (*yellow dots*) and bones (*white line segments*). Mesh parts in *different colors* correspond to different body parts. **c** Triangular mesh for the head (color figure online)

In general, there are differences in body shape and limb lengths between the expert and the performer. Here, we assume that the human body model has been adapted to fit the shape and size of the performer, and the 3-D reference motion has been retargetted to the performer's body before the reference motion and the performer's motion are compared, e.g., using the motion retargetting algorithm in [72]. That is, the human body model (Fig. 1) is that of the performer and the reference motion is retargetted according to performer's body. This is a reasonable assumption because the shapes and sizes of the performer's body can be measured in advance, and retargetting needs to be performed only once for a specific performer. In our application, retargetting adapts the reference motion to the size of the performer.

Note that the reference motion can often be divided into a set of *motion segments* by a set of *segment boundary* frames $\mathcal{T}_b \subset \mathcal{T}$. These reference segment boundaries are determined based on domain knowledge, by which we find that some body parts change their motion directions significantly across segment boundaries. The body part with the most significant change in direction across all segment boundaries is used to analyze the property of segment boundaries (see Sect. 7.1 for more detail). Let $\mathbf{v}_t$ denote the direction of the 3-D velocity of the body part at time t. Then, at segment boundary t, $\mathbf{v}_t \cdot \mathbf{v}_{t+1} < \alpha$, where α is a threshold that depends on the type of motion. For example, if the direction changes at the segment boundaries are greater than 60°, then α can be set $\cos 60° = 0.5$.

3.2 2-D input video

The motion of a performer is captured in the input video, which consists of a sequence of image frames $I'_{t'}$ (Fig. 4a) over time $t' = 0, \ldots, L'$. Typically, $L' < L$ because video camera has a lower sampling rate than 3-D motion capture system. Each input image $I'_{t'}$ contains the image of a performer generated by the projection of an unknown *performer's posture* $B'_{t'}$ onto the image plane. The human body region $S'_{t'}$ in image $I'_{t'}$ is separated from the background using interactive segmentation and skin color detection algorithms [73,74] (Fig. 4b). Note that in a single camera view, depth ambiguity of body parts and self-occlusion can occur.

3.3 3-D–2-D spatiotemporal relationships

There are many complex spatiotemporal relationships between the 3-D reference motion and the 2-D input video. Four major relationships are highlighted below.

1. Temporal difference: The performer's motion can differ from the expert's motion in terms of execution speed. So, a temporal correspondence C needs to be established from 2-D video time t' to 3-D motion time t, i.e., $C(t')$ is a particular t that corresponds to t'. C should satisfy the temporal order constraint: for any two postures in the performer's motion, the two corresponding postures in the reference motion have the same temporal order. Without loss of generality, it is assumed that $C(0) = 0$ and $C(L') = L$.
2. Spatial difference: The performer's (unknown) posture $B'_{t'}$ can differ from the expert's posture $B_{C(t')}$ at the corresponding time frame by a global rigid transformation T and a joint articulation A, i.e., $B'_{t'} = A_{t'}(T_{t'}(B_{C(t')}))$. In the algorithm, $B'_{t'}$ is inferred by registering the projection P of $A_{t'}(T_{t'}(B_{C(t')}))$ to the input body region $S'_{t'}$ in image $I'_{t'}$. Then, the posture error $\varepsilon_{t'}$ is naturally captured in $A_{t'}$ and $T_{t'}$.
3. Smooth motion: The posture error $\varepsilon_{t'}$ can be large when the performer's motion differs significantly from the reference motion. Nevertheless, the rate of change of posture errors should remain small because the motion of interest is smooth. That is, $\Delta\varepsilon_{t'}/\Delta t'$ is small.
4. Segment boundaries: The expert often coaches a performer segment by segment, and pays more attention to the correctness of the beginning and ending postures of each motion segment. When the performer's postures are correct at the boundaries, the postures inside the segment will be more likely correct. This observation implies that errors at the segment boundaries should carry more importance than errors at non-segment boundaries. Therefore, the performer's segment boundaries should match the reference segment boundaries.

3.4 Problem statement

Now, we can formulate the problem of spatiotemporal registration for sports motion analysis as follows:

> Given the reference motion $\{B_t\}$ and the input motion $\{S'_{t'}\}$, determine the temporal correspondence C, projection P, rigid transformation $T_{t'}$, and join articulation $A_{t'}$ that minimize the errors E_S and E_D:
>
> $$E_S = \frac{1}{L'+1} \sum_{t'} d_S \left(P \left(A_{t'} \left(T_{t'} \left(B_{C(t')} \right) \right) \right), S'_{t'} \right), \quad (1)$$
>
> $$E_D = \frac{1}{L'+1} \sum_{t'} \varepsilon_{t'}. \quad (2)$$
>
> E_S is the registration error, where $d_S(S, S')$ is an appropriate difference measure. In our application, edge and silhouette are used as the features for the difference measure d_S which is defined in terms of the amount of overlap between S and S' and the chamfer matching distance between edges extracted from S and S'. The total posture error E_D is minimized to capture the idea of computing the minimum correction required by the performer to match the expert's motion.

The minimization of E_S and E_D is subjected to the following constraints:

A. Joint angle limit. The valid angle between two connected body parts is physically limited to certain ranges.
B. Temporal order constraint. For any t'_1 and t'_2 such that $t'_1 < t'_2$, $C(t'_1) < C(t'_2)$.
C. Small rate of change of posture errors. For each t', $\Delta\varepsilon_{t'}/\Delta t'$ is small.
D. Similarity of corresponding segment boundaries between the reference and the performer's motion. For any segment boundary frame t', $\mathbf{v}_{C(t')} \cdot \mathbf{v}_{C(t'+1)} < \tau$ and $\mathbf{v}'_{t'} \cdot \mathbf{v}'_{t'+1} < \tau$.

When both d_S and $\varepsilon_{t'}$ are minimized, $B'_{t'}$ can be recovered. Consequently, the temporal difference is captured in C and the performer's posture error is measured by $\varepsilon_{t'}$.

4 Spatiotemporal registration framework

It is infeasible to directly solve the proposed problem, which is a very complex high-dimensional optimization problem with long-time sequence. So, it is decomposed into four subproblems and solved in the following stages:

1. Estimation of camera projection P.
   In our application, the performer's motion is assumed to be recorded by a single stationary camera with camera view fixed. The camera projection is assumed to be scaled orthographic because the body movement in depth is in general small compared with the distance from the body to the camera. This stage is omitted in the remainder of this paper in order to focus on the following main stages.
2. Estimation of approximate temporal correspondence C and rigid transformation T.
   Determine initial estimates of C and $T_{t'}$ that minimize the error E_C subject to Constraint B:
   $$E_C = \frac{1}{L'+1} \sum_{t'=0}^{L'} d_S \left(P \left(T_{t'} \left(B_{C(t')} \right) \right), S'_{t'} \right), \quad (3)$$
   Joint articulation $A_{t'}$ is omitted in this stage.
3. Estimation of posture candidates.
   Due to depth ambiguity, multiple postures can match an input body region in the image. So, this stage determines, for each t', multiple $A_{t'l}$ and $T_{t'l}$ that minimize the error $E_{t'}$ subject to Constraint A:
   $$E_{t'} = d_S \left(P \left(A_{t'l} \left(T_{t'l} \left(B_{C(t')} \right) \right) \right), S'_{t'} \right). \quad (4)$$
   The approximate C estimated in the previous stage is used to identify approximate corresponding reference posture $B_{C(t')}$, which is transformed by $A_{t'l}$ and $T_{t'l}$ to match the input body region $S'_{t'}$. This approach avoids the accumulation of estimation error over time, which is present in many human body tracking methods. The resulting $\mathcal{B}_{t'} = \{B'_{t'l}\}$, where $B'_{t'l} = A_{t'l}(T_{t'l}(B_{C(t')}))$ is the set of posture candidates that match $S'_{t'}$ well.
4. Candidate selection and refinement of estimates.
   Select the best posture candidate $B'_{t'}$ from $\mathcal{B}_{t'}$ and determine the C that together minimize E_D subject to Constraints B, C, and D. After finding the best $B'_{t'}$, posture error can be computed as the difference between $B'_{t'}$ and the corresponding $B_{C(t')}$.

The algorithms for Stages 2, 3, and 4 are discussed in the following sections.

5 Estimation of temporal correspondence

This stage estimates approximate temporal correspondence C and transformation T using dynamic programming (DP). The optimal solution at this stage is not globally optimal for the whole problem because articulation is omitted. So, the temporal correspondence estimated at this stage is only an approximation.

Let $d(t', C(t'))$ denote $d_S(P(T_{t'}(B_{C(t')})), S'_{t'})$. The task is to determine C by minimizing E_C:

$$E_C = \frac{1}{L'+1} \sum_{t'=0}^{L'} d(t', C(t')) \quad (5)$$

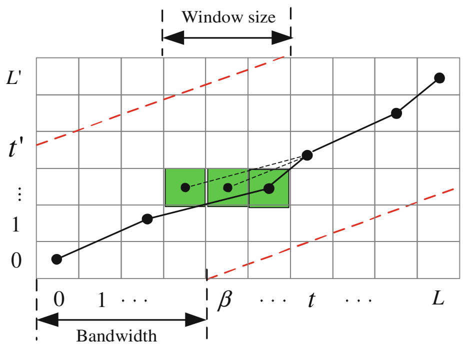


Fig. 2 Correspondence matrix between t' and t. Each *black dot* denotes a correspondence between t' and a unique t, and the *solid line* connecting the *black dots* is a path in the correspondence matrix. The *black dots* in a small window (*green elements*) connected by the *thin dashed lines* denote the possible frame pairs preceding the pair (t', t). The *thick dashed lines* (in *red*) denote the band in which to search for possible correspondences (color figure online)

subject to temporal order constraint. Given a particular C, $T_{t'}$ at each time t' is determined using sampling technique. The DP problem is formulated in the following:

Let $\mathbf{D}$ denote a $(L'+1)\times(L+1)$ correspondence matrix. Each matrix element at (t', t) represents the possible frame correspondence between t' and t, and the correspondence cost is $d(t', t)$. A path in $\mathbf{D}$ is a sequence of frame correspondences for $t' = 0, \ldots, L'$ such that each t' has a unique corresponding $t = C(t')$, with $C(0) = 0$ and $C(L') = L$ (Fig. 2). The cost of a path is the sum of the correspondence costs over all t', and the average path cost is E_C. The problem is to find the least cost path on which E_C is minimized.

The least cost path can be efficiently found by making use of the temporal order constraint. Suppose the frame pair (t', t) is on the least cost path. Then, the possible previous frame pair should be one of $(t'-1, t-1-i)$ for $i = 0, \ldots, w$. The temporal window size w is defined as kL/L' for a small $k \geq 1$. k is small because the change of posture error between the pair of corresponding frames over time is small (Sect. 3.3). The least cost path from the first frame pair $(0, 0)$ to the current pair (t', t) can be determined by recursively computing the least cost path from $(0, 0)$ to one of $(t'-1, t-1-i), i = 0, \ldots, w$.

Let $D(t', t)$ denote the least cost from frame pair $(0, 0)$ up to (t', t) on the least cost path, and $D(0, 0) = d(0, 0)$. Then $D(L', L)$ can be recursively computed as follows:

$$D(t', t) = d(t', t) + \min_{i=0}^{w} D(t'-1, t-1-i) \tag{6}$$

Once $D(L', L)$ is computed, the least cost path is obtained by tracing back the path from $D(L', L)$ to $D(0, 0)$. The least cost path gives the correspondence C (Figs. 2, 5).

Our DP algorithm is similar to dynamic time warping (DTW) [75]. DTW permits one-to-many and many-to-one mappings between t' and t. In addition, two adjacent elements $(t', C(t'))$ and $(t'+1, C(t'+1))$ on the path have to satisfy $C(t'+1) - C(t') \leq 1$. On the other hand, in our DP formulation, each t' corresponds to a unique t (i.e., one-to-one mapping) and $C(t'+1) - C(t') \leq w$.

The computation complexity of DTW is $O(L'L)$, and the complexity of our algorithm is $O(wL'L)$. In the implementation, to improve the efficiency of the algorithm, the possible correspondence can be restricted within a narrow band (Fig. 2, thick dashed lines) along the diagonal of the correspondence matrix because the change of posture error between the pair of corresponding frames over time is small. The bandwidth β of the band is defined as the horizontal distance from the straight diagonal line to the dashed line (Fig. 2). Then the computation complexity of the algorithm is reduced from $O(wL'L)$ to $O(w\beta L')$.

6 Estimation of posture candidates

Posture candidates are estimated using an extension of BP [59,60,46,61,62]. The algorithm uses the approximate temporal correspondence C estimated in the previous stage to identify approximate corresponding reference posture $B_{C(t')}$ at time t' (Fig. 4c). Then, BP uses $B_{C(t')}$ as an initial estimate to search for the posture candidates that match the input body region $S'_{t'}$ (Fig. 4b), thereby determining the candidate articulations $A_{t'l}$ and rigid transformations $T_{t'l}$. In the following, the BP is first described (Sect. 6.1). Then, the nonparametric implementation of BP (Sect. 6.2) and posture estimation algorithm (Sect. 6.3) is developed.

6.1 Belief propagation

Let $p(B'|S')$ denotes the probability that B' is a good posture candidate given input body region S'. Then, posture candidate estimation is to find B' with large $p(B'|S')$. Denote the pose of body part i as b_i, i.e., $B' = \{b_i\}$. Instead of computing $p(B'|S')$ directly, BP iteratively computes $p(b_i|S')$ for each body part i using these equations:

$$p(b_i|S') \propto \phi(b_i, S') \prod_{j\in\Gamma(i)} m_{ji}(b_i) \tag{7}$$

$$m_{ji}(b_i) \propto \int \phi(b_j, S')\; \psi(b_j, b_i) \prod_{k\in\Gamma(j)\backslash i} m_{kj}(b_j)\, db_j \tag{8}$$

where $\Gamma(i)$ is the set of body parts connected to body part i, and $m_{ji}(b_i)$ is the *contribution* of body part j to the pose b_i of body part i. To compute $m_{ji}(b_i)$ and $p(b_i|S')$, the functions $\phi(b_i, S')$ and $\psi(b_i, b_j)$ need to be defined.

The similarity function $\phi(b_i, S')$ measures the degree of match between S' and body part i at pose b_i. Each body part at pose b_i computed in the current iteration is projected and rendered, together with all other body parts whose poses

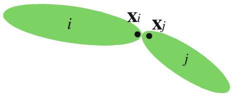


Fig. 3 Joint constraint. The two ends of the connected body parts should be at the same 3-D position, i.e., $\mathbf{x}_i = \mathbf{x}_j$

are obtained in the previous iteration, to produce the projected body region S. Then, the similarity is computed as $\phi(b_i, S') = \exp(-d_S(S, S'))$. $d_S(S, S')$ is defined as above and allows the algorithm to handle partial self-occlusion of body parts. In comparison, the original BP [62] measures similarity using only region overlap between the projection of a single body part and the entire input body region. Therefore, it cannot handle partial self-occlusion of body parts.

The joint constraint function $\psi(b_i, b_j)$ enforces joint constraint and joint angle constraint between two connected body parts i and j. The joint constraint states that two neighboring body parts should be connected at the joint (Fig. 3). Let $\mathbf{x}_i$ and $\mathbf{x}_j$ denote the 3-D positions of the points on body parts i and j that connect to form a joint. When body parts i and j adopt poses b_i and b_j, the degree of satisfaction of joint constraint is measured by $\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/\sigma^2)$, where σ is a positive parameter.

The joint angle constraint ensures that the angle between two connected body parts i and j falls within physical limit. The degree of satisfaction of joint angle constraint is measured by $J(b_i, b_j)$, which is 1 when the joint angle is within limit, and a smaller constant a otherwise. Combining the two constraints, we obtain $\psi(b_i, b_j) = J(b_i, b_j)\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/\sigma^2)$.

The parameters σ and a decrease over iteration. At the first few iterations, the pose estimate of each body part may be far from the actual pose. So, the constraints are loosely enforced initially to ensure that the correct poses can be included. Gradually, the pose estimate of each body part is expected to become more similar to the actual pose, and therefore the constraints should become more strict.

6.2 Nonparametric implementation of belief propagation

In practice, the evaluation of the BP integral in Eq. 8 is often intractable with continuous state variable b_i. Several implementations of BP using nonparametric sampling approach have been proposed [61,60,59]. In this paper, an algorithm similar to BP Monte Carlo [59] is adopted to compute $m_{ji}(b_i)$.

In the algorithm, the possible pose b_i of body part i is represented by a discrete set of samples s_{ilk}, where l denotes the lth sample of the set and k denotes the iteration number of the algorithm. The contribution $m_{ji}(b_i)$ in the kth iteration is represented by the set $\{(s_{ilk}, \omega_{jilk})\}$, where ω_{jilk} is the weight of the contribution from body part j to body part i. The belief $p(b_i|S')$ in the kth iteration is represented by the set $\{(s_{ilk}, \pi_{ilk})\}$, where π_{ilk} is the weight of the belief of body part i. The algorithm iteratively updates the pose of each body part to match the input body region S' in four steps:

Step 1: Decrease parameters
σ and a are gradually decreased over iterations by $\sigma_k = \lambda\sigma_{k-1}$ and $a_k = \{a_{k-1}\}^{1/\lambda}$, where λ is a decreasing factor from 1 to 0. The parameters a is set to decrease at exponential rate of $1/\lambda$ so as to match the influence of σ, which exists in the exponent of the exponential function. A larger λ (e.g., 0.95) can make the beliefs converge to the global optimal estimates with a higher probability. On the other hand, a lower λ (e.g., 0.60) can make the beliefs converge faster, but the beliefs may converge to local optimal estimates with a higher probability. This step is used to harden the joint constraint (Sect. 6.1) to make the beliefs and contributions gradually converge over iterations. Note that there is no such constraint-hardening schedule in [59].

Step 2: Generate new samples
Generate samples s_{ilk} for body part i according to its belief in iteration k and the beliefs of its connected body parts in iteration $k - 1$. The sampling technique consists of three steps:

1. Sample from the nonparametric distributions of belief of body part i and the beliefs of the connected body parts in iteration $k - 1$.
2. For each selected sample, construct a Gaussian function to generate a sample for body part i.
3. Draw a sample from each Gaussian function randomly.

Step 3: Compute the weights of contribution
For each new sample s_{ilk}, compute the weight ω_{jilk} by the nonparametric version of Eq. 8:

$$\omega_{jilk} = \sum_{l'=1} \phi\left(s_{jl',k-1}, S'\right)\psi\left(s_{jl',k-1}, s_{ilk}\right) \times \prod_{h\in\Gamma(j)\setminus i} \omega_{hjl',k-1}. \tag{9}$$

Step 4: Compute the weights of belief
For each new sample s_{ilk}, compute the weight π_{ilk} by the nonparametric version of Eq. 7:

$$\pi_{ilk} = \phi(s_{ilk}, S') \prod_{j\in\Gamma(i)} \omega_{jilk}. \tag{10}$$

The weights π_{ilk} are then normalized such that the sum of them for each body part i is 1:

$$\sum_{l} \pi_{ilk} = 1. \tag{11}$$

The iteration process stops when $p(b_i|S')$ for all i converge or after a fixed number of iterations. The samples with larger weights are the pose estimates of body part i.

In the algorithm, most time is spent on computing the similarity functions $\phi(b_i, S')$. Suppose the number of body

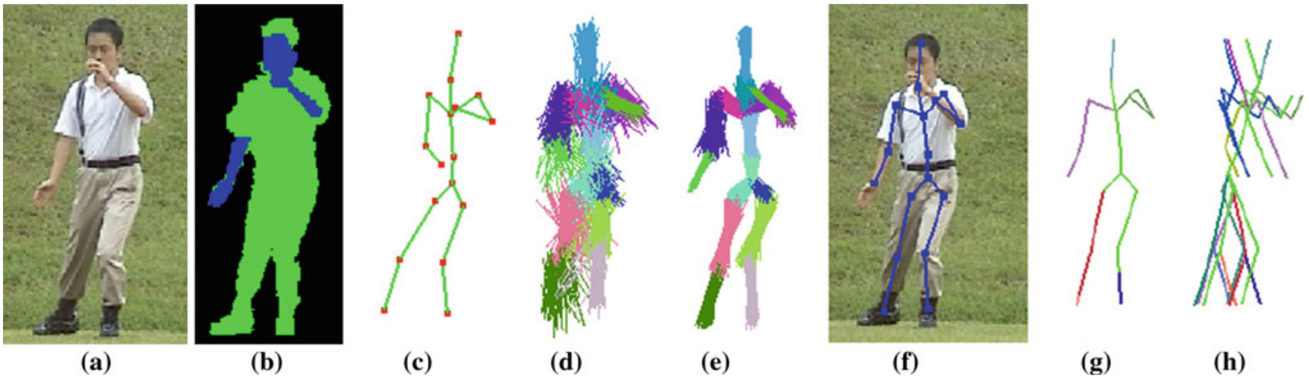


Fig. 4 Estimation of posture candidates. **a** Input image. **b** Input body regions. **c** Approximate corresponding reference posture. **d**, **e** The projections of pose samples of each body part after the 1st and 30th iterations. **f** Posture candidate overlapped onto input image. **g**, **h** Frontal and side views of all posture candidates. Different pose samples and posture candidates are colored with *different colors* (color figure online)

parts is N_i, the number of samples for each body part is N_l, and the number of iterations is N_k. Then, the computation complexity of the algorithm is $\mathrm{O}(N_i N_l N_k)$.

Note that if a body part is totally occluded, the belief of the part will be influenced by several factors: the beliefs of connected body parts, the corresponding reference posture, and the posture candidate estimated in the previous frame. If the neighboring body parts are close to the ground truth, the joint constraints between the neighboring parts and the current part will generate pose samples of the current body part that are close to the ground truth. Similarly, if the corresponding reference posture and the posture candidates in the previous frame are similar to the performer's posture in the current frame, the pose samples of the current body part will be close to its true belief. Otherwise, the pose estimate may be quite different from the ground truth.

6.3 Posture candidate estimation algorithm

The BP algorithm described earlier estimates only the pose samples of each body part (Fig. 4d, e). These pose samples are used to generate posture candidates as follows. The first posture candidate is computed such that each body part has the same depth orientation as that in the corresponding reference posture, and its projection matches the mean of its pose samples (Fig. 4f). Then, based on the first posture candidate, flip the depth orientation of n body parts about their parent joints, starting with $n = 1$, while keeping the body parts connected at the joints. This step is repeated for $n = 1, 2, \ldots,$ until enough (experimentally 20 to 50) posture candidates are generated. These posture candidates have exactly the same frontal projection (Fig. 4g), but different side projections (Fig. 4h). Therefore, they capture all possible depth ambiguities in the image of a single camera view.

7 Refinement of estimates

This stage selects the best posture candidate at each t' that minimize the error E_D (Eq. 2), and simultaneously refines the temporal correspondence C, subject to Constraints B, C, and D. To satisfy the segment boundary constraint (Constraint D), $\sum_{t'} \varepsilon_{t'}$ needs to be minimized within each motion segment. Therefore, it is necessary to identify the performer's segment boundaries in the performer's motion given the reference segment boundaries in the reference motion.

7.1 Determination of performer's segment boundaries

Given the set $\mathcal{T}_b$ of reference segment boundaries that are known in advance, the approximate temporal correspondence C, and the posture candidates $B'_{t'l'}$ at each t', the objective is to to determine the performer's segment boundaries in the performer's motion.

For each reference segment boundary $t \in \mathcal{T}_b$, the corresponding performer's segment boundary is determined by the following steps:

1. Obtain a temporal window $[t' - \omega, t' + \omega]$, where ω is the window size, and t' is the initial estimate of the performer's segment boundary determined by $C(t') = t$.
2. Find one or more smooth sequences of posture candidates in the temporal window.
   - Correct posture candidates should change smoothly over time. Suppose $B'_{\tau l'}$ and $B'_{\tau+1,k'}$ are correct posture candidates, then $d_B(B'_{\tau l'}, B'_{\tau+1,k'})$ is small for any $\tau \in [t' - \omega, t' + \omega]$.
   - Choose a posture candidate for each $\tau \in [t' - \omega, t' + \omega]$ to obtain a sequence of posture candidates that satisfy the condition that $d_B(B'_{\tau l'}, B'_{\tau+1,k'})$ is small for each τ.

3. Find candidate segment boundaries.
   - For each smooth sequence of posture candidates, find the candidate segment boundary $\tau \in [t' - \omega, t' + \omega]$ and the corresponding posture candidate at τ that satisfies the segment boundary condition (Sect. 3.4).
   - Denote a candidate segment boundary found above as τ_i and the corresponding posture candidate as B'_i.
4. Identify the optimal segment boundary τ^*.
   The posture candidate at the optimal segment boundary τ^* should be the most similar to the corresponding reference posture B_t. Therefore, τ^* can be determined as follows:

$$\tau^* = \tau_k \,, \quad k = \arg\min_i d_B(B_t, B'_i). \tag{12}$$

7.2 Refinement of estimates within each motion segment

After determining the performer's segment boundaries, a posture candidate has to be selected at each t' within each motion segment to determine the optimal C and compute the posture errors. Let $[t_b, t_e]$ denotes a reference motion segment and $[t'_b, t'_e]$ denotes the corresponding performer's motion segment. Let $\ell(t')$ denotes the index of the best posture candidate at t' within the motion segment $[t'_b, t'_e]$. Then, the problem is to determine the ℓ and C that minimize E_D subject to Constraints B and C. Constraint C, i.e., small rate of change of posture errors, can be incorporated into E_D to obtain E_F:

$$E_F = \frac{1}{t'_e - t'_b + 1} \sum_{t'=t'_b}^{t'_e} [d_c(t', C(t'), \ell(t')) \\ + \lambda\, d_s(t', C(t'), C(t'-1), \ell(t'), \ell(t'-1))], \tag{13}$$

where λ is a weighting factor. The difference d_c is obtained from E_D, i.e., $d_c(t', t, l') = \varepsilon_{t'} = d_B(B_t, B'_{t'l'})$. d_B is the posture error between the posture candidate $B'_{t'l'}$ and the reference posture B_t, which is defined as the mean orientation difference of all body parts in the postures. The difference $d_s(t', t, s, l', k')$ measures the change of posture errors between two pairs of corresponding postures $(B'_{t'l'}, B_t)$ and $(B'_{t'-1,k'}, B_s)$:

$$d_s(t', t, s, l', k') = \left[d_B(B_t, B'_{t'l'}) - d_B(B_s, B'_{t'-1,k'}) \right]^2. \tag{14}$$

DP technique similar to that in Sect. 5 is developed to determine the optimal $\ell(t')$ and $C(t')$. In this case, the correspondence matrix $\mathbf{D}$ is a $(t'_e - t'_b + 1) \times (t_e - t_b + 1) \times N_B$ matrix, where N_B is the maximum number of posture candidates at each t'. Each matrix element at (t', t, l') represents the possible correspondence between posture candidate $B'_{t'l'}$ and reference posture B_t. The correspondence cost consists of two terms: $d_c(t', t, l')$ and $d_s(t', t, s, l', k')$. A path in $\mathbf{D}$ is a sequence of correspondences for $t' = t'_b, \ldots, t'_e$ such that each t' has a unique corresponding $t = C(t')$ and $l' = \ell(t')$. The cost of a path is the sum of the correspondence costs over all t', and the average path cost is E_F. The problem is to find the least cost path on which E_F is minimized.

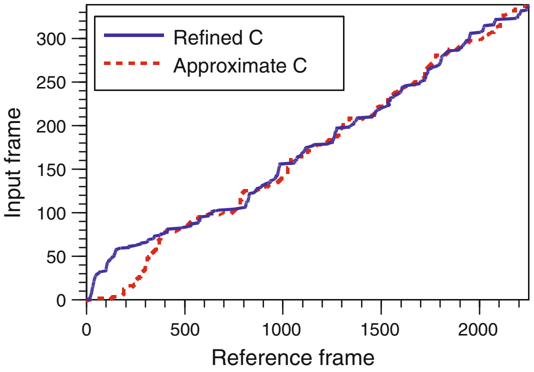


Fig. 5 Approximate and refined temporal correspondence C

Let $D(t', t, l')$ denotes the least cost from the triplet (t'_b, t_b, l'_b) up to (t', t, l') on the least cost path, and therefore $D(t'_b, t_b, l'_b)$ is $d_c(t'_b, t_b, l'_b)$. Then, by a similar reasoning as in Sect. 5, $D(t'_e, t_e, \ell(t'_e))$ can be computed recursively using the formulae

$$D(t', t, \ell(t')) = \min_{l'} D(t', t, l') \tag{15}$$

$$\ell(t') = \arg\min_{l'} D(t', t, l') \tag{16}$$

$$D(t', t, l') = d_c(t', t, l') + \min_{i,k'} \{D(t'-1, t-1-i, k') \\ + d_s(t', t, t-1-i, l', k')\}. \tag{17}$$

Once $D(t'_e, t_e, \ell(t'_e))$ is computed, the least cost path can be obtained by tracing back the path from $D(t'_e, t_e, \ell(t'_e))$ to $D(t'_b, t_b, \ell(t'_b))$. Test result in Fig. 5 shows that the refined optimal C is not a linear function.

8 Experiments and discussions

We split the test into two phases. First, synthetic data were mainly used to assess the accuracy of the posture candidate estimation algorithm. The test results gave an estimate of the algorithmic error in estimating the performer's actual 3-D postures from 2-D input images. Next, the whole spatiotemporal registration algorithm was tested on real data to measure the performer's posture error. As long as the measured error is significantly greater than the algorithmic error, we are confident that the measured error reliably reflects the actual posture error of the performer. Two sets of motion sequences were mainly used for the tests: (1) 3-D Taichi reference motion with 2,250 reference postures captured by a commercial motion capture system, and input video with

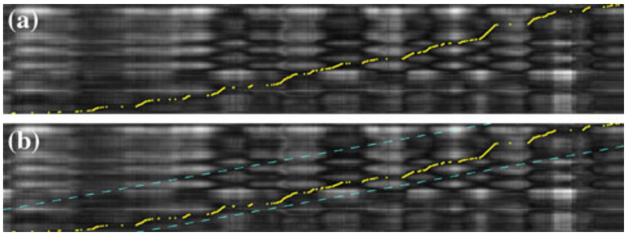


Fig. 6 Approximate temporal correspondence. **a** The optimal approximate C shown as the *yellow dots* along the diagonal. **b** Approximate C with $k = 10$ and $\beta/L = 20\%$, which is almost as same as the optimal C in (**a**) (color figure online)

339 input images of size 320 × 240, and (2) 3-D golf swing motion with 250 reference postures and input video with 51 input images.

8.1 Estimation of approximate temporal correspondence

This test evaluates the performance of the algorithm for estimating the approximate temporal correspondence. An input video consisting of 399 images was used to determine the window size parameter k and bandwidth β. Figure 6a illustrates a visualization of the correspondence matrix and the optimal approximate temporal correspondence with the maximum normalized bandwidth $\beta/L = 100\%$ and $w = L$. The intensity of pixel (t, t') represents the difference $d(t', t)$. So, a darker pixel represents smaller difference. Note that the pixel $(0, 0)$ is located at the bottom-left, and the original $d(t', t)$ value is scaled to intensity value between 0 and 255.

From tests, we found that the optimal solution of the approximate temporal correspondence can be obtained for window size $k \in [10, L']$ and bandwidth $\beta/L \in [20\%, 100\%]$. That means, a small window size and bandwidth are enough for finding the optimal solution. By setting $k = 10$ and $\beta/L = 20\%$, the least cost path in Fig. 6b is almost as same as the optimal solution. At these settings, 60% of the computation time is saved compared with searching the correspondence matrix with window size $k = 10$ and $\beta/L = 100\%$. Figure 7 illustrates several pairs of input images and their corresponding reference postures obtained based on the approximate temporal correspondence. We can see that the corresponding reference postures are indeed similar to the performer's postures in the input images.

8.2 Accuracy of posture error estimation

In this test, synthetic test data were generated as follows. One hundred and ten reference postures were selected at regular intervals from the 3-D Taichi sequence. Each selected 3-D posture was mapped to an articulated 3-D human model, which was projected by scaled orthographic projection and rendered using OpenGL to obtain a synthetic input image.

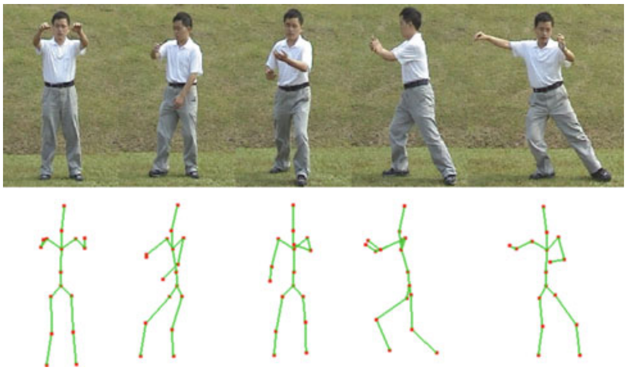


Fig. 7 Examples of input images (*first row*) and their corresponding reference postures (*second row*)

The 3-D reference posture served as the ground truth of the input image. Next, the joint angles of the ground truth posture were changed by random values in the range [−20°, +20°] to generate a new posture to serve as the initial posture for the posture candidate estimation algorithm. This approach was adopted to emulate the real application situation that the actual performer's posture may differ from the initial posture estimate. Note that some of the synthetic input images generated contained self-occlusion and depth ambiguity. In the experiments, the decreasing factor λ was set to 0.9, and the number of iterations was set to 40. In each iteration of the algorithm, 300 pose samples were generated for each body part. About 8 s were spent for each iteration, most of which were used to compute the similarity function.

Figure 8 illustrates a sample test result. Given the input image (Fig. 8b) and the initial posture (Fig. 8c) generated by articulating the ground truth 3-D reference posture (Fig. 8a), multiple posture candidates were estimated by the posture estimation algorithm. All posture candidates had the same projections from the frontal view (Fig. 8d), but they differed in depth orientations for some body parts, as revealed in the side views (Fig. 8e). Figure 8f illustrates the side view of the best posture candidate in the candidate set. From Fig. 8f and d, we can see that the best posture candidate is very similar to the ground truth (Fig. 8a).

For the sample input image (Fig. 8b) and the initial posture (Fig. 8c), Fig. 9 shows the decreasing trend of 2-D joint position error E_{2P} with respect to the iteration number of the BP algorithm, where E_{2P} is the mean error of all body joints' image positions projected by the posture candidates with respect to the ground truth. The error decreases to a small value of about 1 pixel after 35 iterations, which indicates that the posture estimation algorithm converges after 35 iterations. A 2-D joint position error of 1 pixel is the best that can be achieved without using sub-pixel algorithm.

Among the posture candidates (Fig. 8d, e), there is one best candidate (Fig. 8e) that is most similar to the ground truth. For the algorithm to be accurate, the posture error

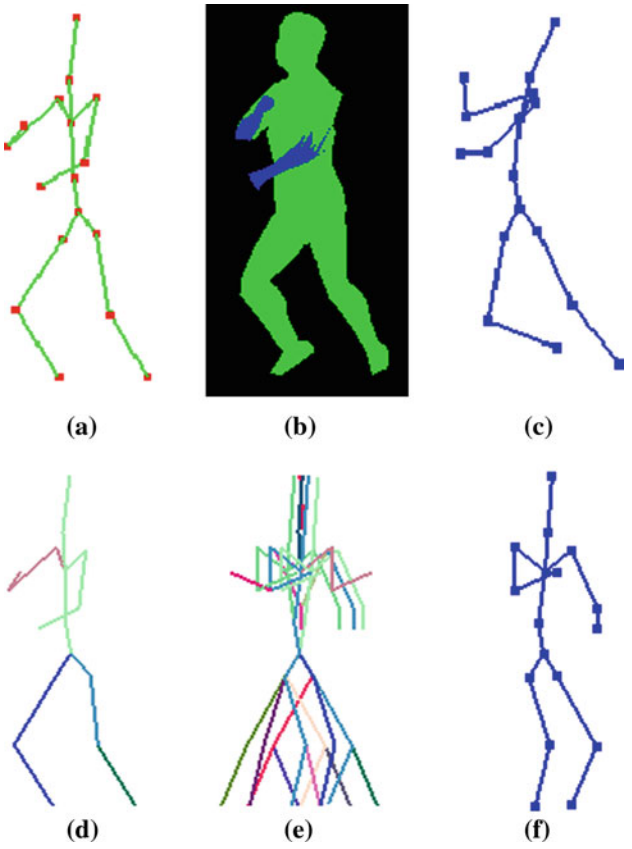


Fig. 8 Posture candidate estimation from a synthetic input image. **a** Ground truth 3-D posture. **b** Synthetic input image. **c** Initial posture. **d**, **e** Posture candidates viewed from the front and the side. **f** Best posture candidate viewed from the side

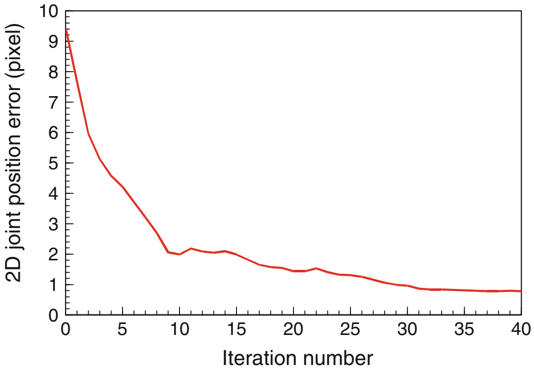


Fig. 9 2-D joint position error E_{2P} with respect to iteration number

between the best candidate and the ground truth should be small. Figure 10 illustrates the posture errors of the posture candidates that best match the ground truth postures for all input images. It shows that the algorithmic error ranges from 2° to 15°, with a mean of 7° and a standard deviation of 2.6°. The larger errors occur in the input images with total occlusion of some body parts. For the other input images, the errors are mainly due to depth ambiguity of body parts. In our test, a body part of length 30 cm parallel to the image plane measures about 36 pixels in the image, and the length of the body part in the image changes by only one pixel when the body part is rotated by 14° in depth. Therefore, a mean error of 7° is reasonable and acceptable for an algorithm that uses a single camera view. The accuracy can be further improved using images with larger resolution or sub-pixel algorithm, which will take more time.

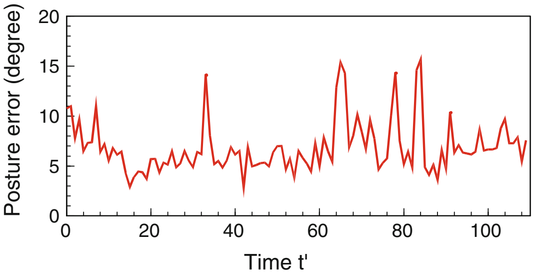


Fig. 10 Algorithmic error in estimating performer's posture

Figure 11 shows sample test results for the synthetic images. From the second row, we can see that all estimated posture candidates are aligned when viewed from the front. That is, their projections match the input image equally well. However, due to depth ambiguity, these posture candidates are not the same, as revealed in their side views (third row). From the fourth row, we can see that the best posture candidate for each input image corresponds to the ground truth posture in the sense that the depth orientation of each body part in the best candidate is as same as that in the ground truth.

Table 1 shows a comparison with a state-of-art 2-D posture estimation method [76] ('YR' in Table 1) based on the real Taichi video. One hundred and thirteen input images were regularly sampled from the video, and the ground truth 2-D position of each body joint was manually annotated for each sampled image. For the YR method [76], the full-body model was trained on the Image Parse dataset [51] and then applied to estimate 2-D positions of each joint over all the sampled images. Note that the proposed method in this paper relies on background subtraction, while the YR method estimates 2-D postures in the whole image regions. To make a relatively fair comparison, only the correctly located joints by the YR method were used to compute the 2-D position errors, where a joint is considered correctly localized if the joint estimate is within 50% of the ground truth length of the related body part from its true position (as in [76]). In this case, the proposed method gave either significantly better or comparable estimates (Table 1) for all body joints.

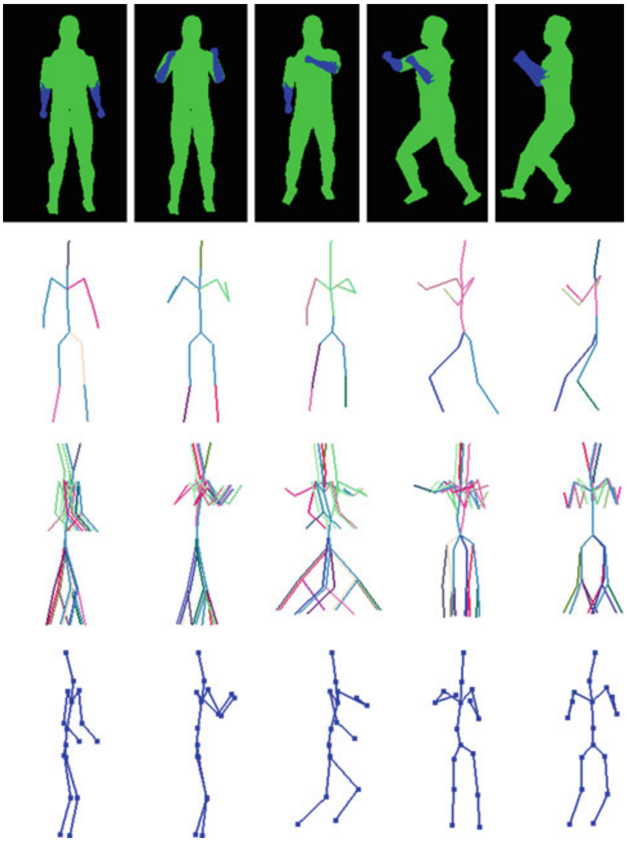


Fig. 11 Estimation of posture candidates from synthetic images. *First row* displays input images. *Second* to *third rows* displays the frontal and side views of all posture candidates for each input image. Every candidate has a unique *color*. *Fourth row* displays the best posture candidates viewed from the side (color figure online)

8.3 Estimation of performer's segment boundaries

In this test, the performance of the algorithm for estimating the performer's segment boundaries was evaluated. First, 25 posture candidates were estimated by running the algorithms in the previous stages (Sect. 6). Then, the performer's segment boundaries were estimated by the segment boundary estimation algorithm. At the reference segment boundaries (RSB, second row in Table 2), there is always a large direction change for the right wrist by at least 60° (Fig. 12). Therefore, the right wrist was used as the joint that indicates segment boundary and the direction change threshold was set at 60° for the Taichi motion. For golf swing motion, the threshold was found to be 120°.

Table 2 Segment boundaries of Taichi sequence

SB	B1	B2	B3	B4	B5	B6	B7
RSB	0	215	570	972	1,535	1,733	2,249
PSBG	0	60	91	145	227	262	338
PSBI	0	16	85	134	233	265	338
PSBF1	0	60	90	143	228	261	338
PSBF2							
Actual	0	29	46	70	114	131	169
Up-scaled	0	58	92	140	228	262	338

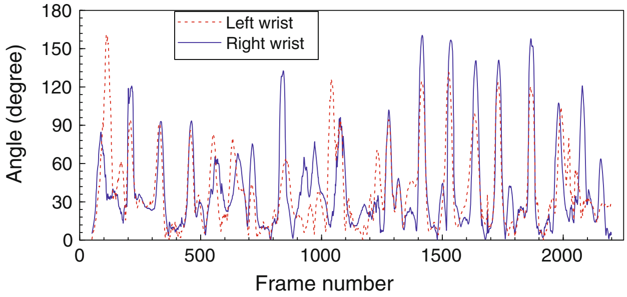


Fig. 12 The change of motion direction for the left wrist and the right wrist joints in the Taichi reference motion. Not all frames with large direction change correspond to the reference boundaries because the hands can change motion directions more than once in a motion segment

Table 2 illustrates the test results. From the fourth row, we can see that the initial estimates (PSBI) are very different from the ground truth performer's segment boundaries (PSBG) because the temporal correspondence determined at Stage 2 is only an approximate that does not take into account articulation of body parts. In comparison, the final estimates of the performer's segment boundaries (PSBF1, fifth row) differ from the ground truth (third row) by at most two frames, which is reasonably small in an input video of 339 frames. When the input video has a higher frame rate (50 fps), the estimates of the performer's segment boundaries (last row) are also very close to the ground truth, which indicates that the algorithm is robust. The frame numbers of the performer's segment boundaries for 50 fps input video are up-scaled for ease of comparison with the ground truth.

Figure 13 visually shows some of the boundary estimation results. From the results, we can see that the performer's postures in all displayed input images are similar to the

Table 1 2-D position errors (pixels) for different joints: average errors and standard deviations (in *brackets*)

Method	RA	RK	RH	LH	LK	LA	RW	RE	RS	LS	LE	LW
Our method	2.0 (1.3)	5.3 (2.8)	3.1 (1.6)	4.3 (2.8)	6.1 (3.6)	3.1 (1.9)	4.6 (3.2)	4.5 (3.9)	4.8 (3.8)	4.4 (2.2)	2.7 (1.5)	2.8 (1.9)
YR [76]	5.6 (3.1)	5.3 (3.4)	11.7 (3.6)	10.6 (4.4)	5.2 (3.0)	6.1 (3.0)	6.1 (2.8)	5.6 (2.7)	5.2 (2.5)	4.9 (2.1)	4.9 (2.8)	7.1 (3.4)

R right, *L* left, *A* ankle, *K* knee, *H* hip, *W* wrist, *E* elbow, *S* shoulder

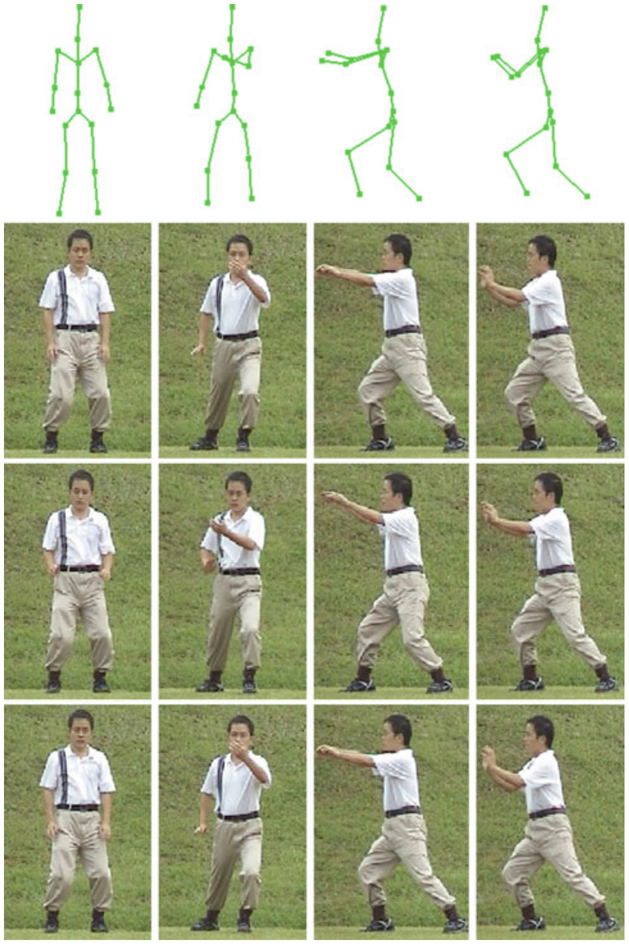


Fig. 13 Visual illustrations of segment boundaries. *First row* displays reference postures at the reference segment boundaries. *Second* to *fourth rows* displays the input images of the ground truth, the initial estimates, and the final estimates at the performer's segment boundaries

corresponding reference postures. It indicates that the performer's segment boundaries cannot be accurately estimated only by the similarity between the input body regions and the projected body region at the reference segment boundaries. This also explains why the initial estimates are not accurate because the approximate temporal correspondence is determined without taking into account articulation of body parts. In contrast, the segment boundary estimation algorithm accurately estimates the performer's segment boundaries by using the segment boundary properties (Sect. 3.3).

8.4 Reliability of posture error estimation

In this test, the spatiotemporal registration algorithm was executed on the Taichi sequence and the golf sequence. Then, the posture error between the selected best posture candidate and the corresponding reference posture was computed for each input image in the motion sequences.

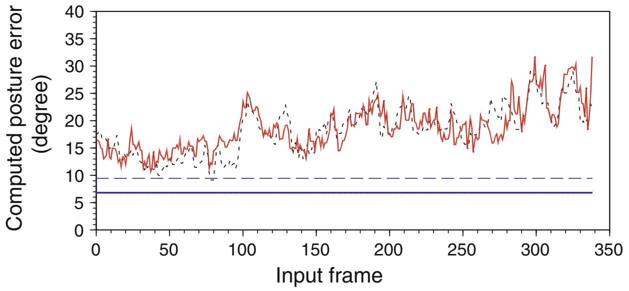


Fig. 14 Computed posture error for Taichi motion. *Solid curve* indicates computed posture error. *Dotted curve* indicates computed posture error in the inverse time order. *Solid line* indicates algorithmic error in estimating postures in synthetic data. *Dashed line* indicates expected algorithmic error

Figure 14 illustrates the computed errors for the Taichi sequence. As discussed in the previous section, the algorithm has a mean error of 7° (solid line in Fig. 14) in estimating postures in synthetic data. For real images, this algorithmic error is expected to be larger, say the mean error plus the standard deviation (dashed line in Fig. 14). The computed error includes both algorithmic error and performer's actual posture error. Since the algorithmic error is small compared with the computed error, there is high confidence that the computed error indeed reflects the performer's error. In addition, in order to testify that the algorithm can avoid the accumulation of estimation error over time, posture candidates are estimated in the reverse time order, i.e., starting from the last frame and ending to the first frame. In this case, the corresponding posture errors (dotted curve in Fig. 14) are shown to have similar error compared with the solid curve, which indicates that there is no accumulation error in the algorithm.

Figure 14 also shows that the computed posture errors are relatively small in most of the first 100 frames compared with the later frames. This is reasonable because the performer started from a standard standing posture which was easy to perform correctly. As the performer moved on to the more difficult postures, more error were made.

Figure 15 shows sample results of the Taichi sequence with small posture errors. The selected posture candidates are similar to the corresponding reference postures. The depth orientations of the body parts in the selected posture candidates are as same as those in the performer's postures in the input images. These results qualitatively verifies that the algorithm can select the best posture candidates.

Figure 16 shows sample results of the Taichi sequence with larger posture errors. Comparing the best posture candidates selected by the algorithm (blue) with the corresponding reference postures (green), there are large errors in the poses of the performer's arms. It shows that the algorithm can indeed identify errors in the performer's postures.

Figure 17 illustrates posture error results for the golf swing motion. Similar to the Taichi case, the performer made less

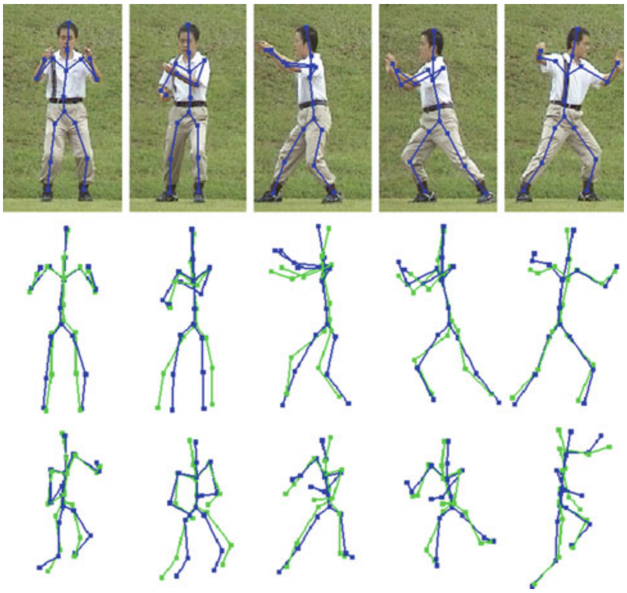


Fig. 15 Sample postures in Taichi sequence with small errors. *First row* displays input images with the selected posture candidates overlaid. *Second* and *third rows* displays selected posture candidates (*blue skeleton*) overlapped with the corresponding reference postures (*green skeleton*) in the frontal and oblique views (color figure online)

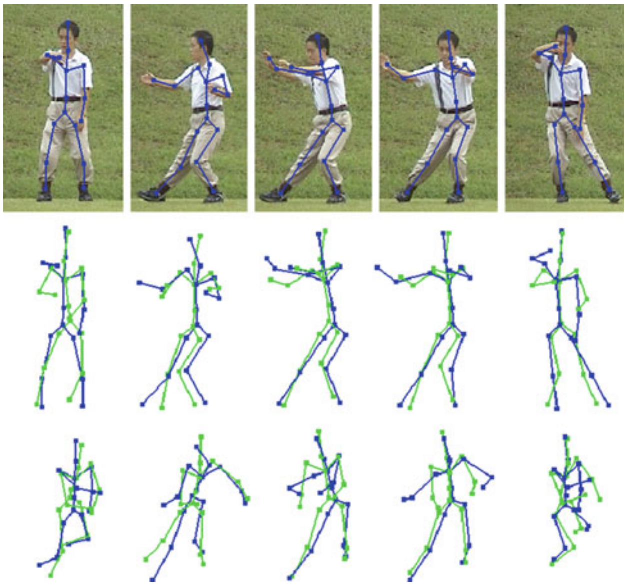


Fig. 16 Sample postures in Taichi sequence with larger errors

error at the beginning of the swing and larger error later on in the swing. This is visually confirmed by the sample results illustrated in Fig. 18. The depth orientations of body parts in the selected posture candidates are as same as those in the performer's posture in the input images. These results verify that the algorithm can be applied to the analysis of different types of sports motion.

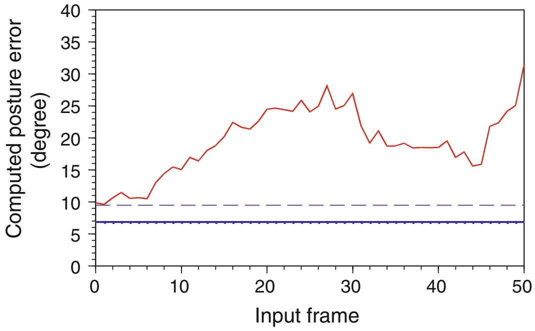


Fig. 17 Computed posture error for golf swing

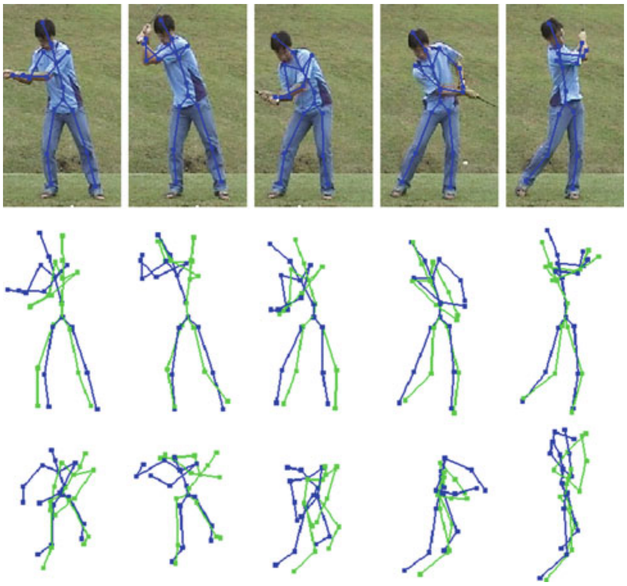


Fig. 18 Sample postures in golf swing sequence

8.5 Robustness of posture error estimation

Figure 19 shows sample test results of the Taichi sequence under ambiguous conditions. Depth ambiguity exists in all images and self-occlusion of the right arm exists in the last three images. Nevertheless, the algorithm can still infer the pose of the occluded body part when the performer's posture does not differ greatly from the reference posture. That is, the algorithm is robust against depth ambiguity and self-occlusion. Of course, when the pose of the totally occluded body part differs significantly from that in the reference posture, no information will exist in a single camera view for the algorithm to infer the actual pose.

9 Conclusions

This paper proposes a novel and fundamental problem for sports motion analysis: 3-D–2-D spatiotemporal motion

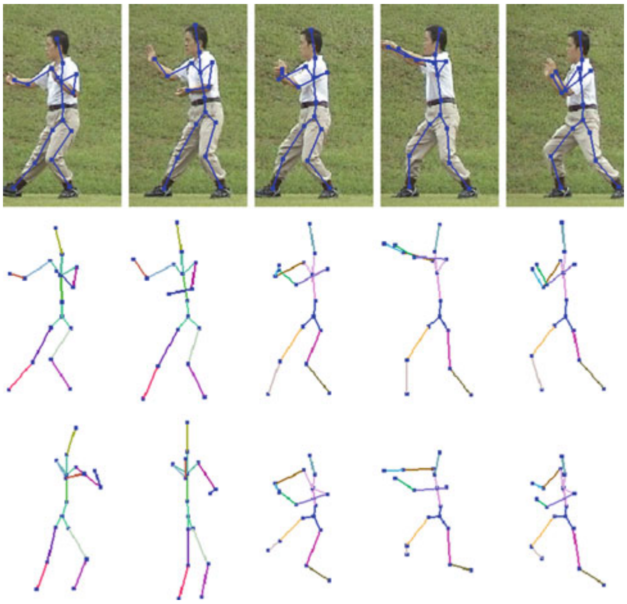


Fig. 19 Selected best postures under ambiguous conditions. *row 3* The first two images show the side views of selected posture candidates, and the last three show the frontal oblique views. Each bone is marked with a unique *color* for easy identification (color figure online)

registration. Since it is infeasible to directly solve such a complex problem, this paper presents a framework that decomposes the problem into four subproblems, which are solved in stages. By using reference postures as initial postures to estimate possible posture candidates in the input images, the algorithm avoids the accumulation of estimation error over time. Moreover, the algorithm seeks to compute the smallest amount of correction required by the performer to match the reference motion. Comprehensive tests were performed to evaluate the performance of the algorithms. Test results show that the computed errors are significantly larger than the expected algorithmic errors when performer's errors occur. This indicates that there is high confidence that the computed errors indeed reflect the performer's errors. In addition, the algorithm can handle depth ambiguity and partial self-occlusion of body parts. In the case of total self-occlusion, the algorithm can infer the pose of the occluded body part if the performer's posture does not differ greatly from the reference posture. The algorithm can also be applied to analyze different types of sports motion, which has been demonstrated using Taichi and golf swing motion. As part of future work, more types of sport motion can be used to test the performance of the proposed framework and algorithms.

10 Limitations and future work

The algorithm works under certain reasonable assumptions. It assumes that the user performs the same type of sport motion as the expert, and therefore the performer's motion is more or less similar to the expert's reference motion. The human body model is assumed to be that of the performer and the reference motion is retargetted according to performer's body. It also assumes that the performer's motion is recorded by a single stationary camera with camera view fixed. The camera projection is assumed to be scaled orthographic because the body movement in depth is small compared with the distance from the body to the camera in our application. Perspective camera model can be used to improve the accuracy of camera projection. Furthermore, the background is assumed to be static and different from the foreground (i.e., human body region) in color in order robustly segment out the human body region from each image. Images with dynamic or cluttered background will make the foreground–background segmentation difficult.

Besides the above assumptions, the limitations and possible solutions of the motion analysis framework are described in the following. The implementation of the possible solutions are considered to be part of future work.

First, the current implementation of the framework works only when the input video and the reference motion begin and end at the corresponding frames, i.e., $C(0) = 0$ and $C(L') = L$. When this condition is not satisfied, dynamic programming can be applied in Stage 2 to determine the best temporal correspondence within a window in the reference motion for the beginning and the end of the input video.

Second, when some body parts are totally occluded in the input images, their poses are unknown and the estimated pose samples may be quite far from the ground truth. This problem can be controlled by checking whether a body part is occluded when it is projected to 2-D image plane during pose estimation. If it is occluded, the pose sample can be replaced by the one in the reference posture such that the projected 2-D joint position error will not be arbitrarily large. Multiple cameras can also be used to efficiently resolve the occlusion problem by recording the performer's motion from multiple views.

Third, in practise, the user may forget to perform a motion segment or repeat some motion segments incorrectly. In this case, to obtain an optimal temporal correspondence between the performer's motion and the reference motion, the missing or extraneous motion segments should be determined. One possible way for determining such segments is to find all segment boundary candidates in the performer's motion, and then use dynamic programming to determine the correct correspondence between the performer's segment boundary candidates and the reference segment boundaries.

When the performer's posture is very different from the corresponding reference posture, the reference posture cannot provide a good initial estimate for posture estimation. In this case, the estimated posture candidates in the previous frame can be used as the initial estimates, or the belief propagation method can be replaced by, e.g., mapping function-based methods which require no initial estimates.

In addition, the computed posture error can be mapped to domain-specific error based on the domain knowledge so that feedback to the performer is more useful and direct in improving his motion. For example, the torso should be upright in most Taichi postures. So, a small error in torso orientation is considered as a major error by the domain-specific criteria. On the other hand, some posture errors are not important for computing the domain-specific error. For example, in Taichi, the knee's joint angle is allowed to vary according to whether the performer is practicing "high stance" or "low stance".

All future works would make the system more feasible for real practical applications.

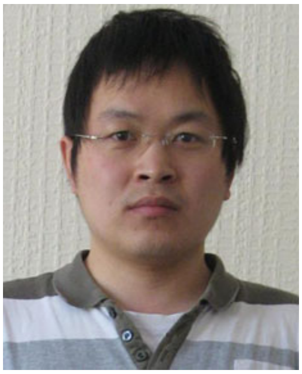
References

1. Simi: 3D motion tracking system. http://www.simi.com
2. Vicon: Optical motion capture system. http://www.vicon.com
3. MotionCoach: Golf swing analysis. http://www.motioncoach.com
4. Simi: Video based motion analysis. http://www.simi.com
5. Sports Motion: 2D video-based motion analysis system. http://www.sports-motion.com
6. V1 Pro: Golf swing analysis software. http://www.ifrontiers.com
7. Bregler C., Malik J.: Tracking people with twists and exponential maps. In: Proc. CVPR, pp. 8–15 (1998)
8. Sidenbladh H., Black M., Fleet D.: Stochastic tracking of 3D human figures using 2D image motion. In: Proc. ECCV, pp. 702–718 (2000)
9. Sminchisescu C., Triggs B.: Kinematic jump processes for monocular 3D human tracking. In: Proc. CVPR, pp. 69–76 (2003)
10. Li R., Yang M.H., Sclaroff S., Tian T.P.: Monocular tracking of 3D human motion with a coordinated mixture of factor analyzers. In: Proc. ECCV, pp. 137–150 (2006)
11. Sminchisescu C., Triggs B.: Covariance scaled sampling for monocular 3D body tracking. In: Proc. CVPR, pp. 447–454 (2001)
12. Urtasun R., Fleet D.J., Fua P.: 3D people tracking with gaussian process dynamical models. In: Proc. CVPR, pp. 238–245 (2006)
13. Capsi, Y., Irani, M.: Spatio-temporal alignment of sequences. IEEE Trans. PAMI **24**(11), 1409–1424 (2002)
14. Rao C., Gritai A., Shah M., Mahmood T.S.: View-invariant alignment and matching of video sequences. In: Proc. ICCV, pp. 939–945 (2003)
15. Moeslund, T.B., Hilton, A., Kruger, V.: A survey of advances in vision-based human motion capture and analysis. Computer Vis. Image Underst. **104**(2), 90–126 (2006)
16. Agarwal A., Triggs B.: 3D human pose from silhouettes by relevance vector regression. In: Proc. CVPR, pp. 882–888 (2004)
17. Bissacco A., Yang M.H., Soatto S.: Fast human pose estimation using appearance and motion via multi-dimensional boosting regression. In: Proc. CVPR (2007)
18. Elgammal A., Lee C.S.: Inferring 3D body pose from silhouettes using activity manifold learning. In: Proc. CVPR, pp. 681–688 (2004)
19. Fossati A., Salzmann M., Fua P.: Observable subspaces for 3D human motion recovery. In: Proc. CVPR (2009)
20. Ning H., Xu W., Gong Y., Huang T.: Discriminative learning of visual words for 3D human pose estimation. In: Proc. CVPR (2008)
21. Rosales R., Athitsos V., Sclaroff S.: 3D hand pose reconstruction using specialized mappings. In: Proc. ICCV, pp. 378–385 (2001)
22. Rosales R., Sclaroff S.: Specialized mappings and the estimation of human body pose from a single image. In: Workshop on human motion, pp. 19–24 (2000)
23. Sminchisescu C., Kanaujia A., Li Z., Metaxas D.: Discriminative density propagation for 3D human motion estimation. In: Proc. CVPR, pp. 390–397 (2005)
24. Thayananthan A., Navaratnam R., Stenger B., Torr P.H.S., Cipolla R.: Multivariate relevance vector machines for tracking. In: Proc. ECCV, pp. 124–138 (2006)
25. Urtasun R., Darrell T.: Sparse probabilistic regression for activity-independent human pose inference. In: Proc. CVPR (2008)
26. Tipping M.: The relevance vector machine. In: NIPS (2000)
27. Lee C.S., Elgammal A.: Modeling view and posture manifolds for tracking. In: Proc. ICCV (2007)
28. Navaratnam R., Fitzgibbon A., Cipolla R.: The joint manifold model for semi-supervised multi-valued regression. In: Proc. ICCV (2007)
29. Athitsos V., Alon J., Sclaroff S., Kollios G.: Boostmap: A method for efficient approximate similarity rankings. In: Proc. CVPR, pp. 268–275 (2004)
30. Athitsos V., Sclaroff S.: Inferring body pose without tracking body parts. In: Proc. CVPR, pp. 721–727 (2000)
31. Athitsos V., Sclaroff S.: Estimating 3D hand pose from a cluttered image. In: Proc. CVPR (2003)
32. Faloutsos C., Lin K.I.: Fastmap: a fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In: ACM SIGMOD, pp. 163–174 (1995)
33. Hjaltason, G.R., Samet, H.: Properties of embedding methods for similarity searching in metric spaces. IEEE Trans. PAMI **25**(5), 530–549 (2003)
34. Howe N.R.: Silhouette lookup for automatic pose tracking. In: CVPR Workshop, pp. 15–22 (2004)
35. Mori G., Malik J.: Estimating human body configurations using shape context matching. In: Proc. ECCV, pp. 666–680 (2002)
36. Shakhnarovich G., Viola P., Darrell T.: Fast pose estimation with parameter-sensitive hashing. In: Proc. ICCV, pp. 750–757 (2003)
37. Roweis, S., Saul, L.: Nonlinear dimensionality reduction by locally linear embedding. Science **290**(5500), 2323–2326 (2000)
38. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science **290**, 2319–2323 (2000)
39. Andriluka M., Roth S., Schiele B.: Pictorial structures revisited: people detection and articulated pose estimation. In: Proc. CVPR (2009)
40. Ferrari V., Marin-Jimenez M., Zisserman A.: Progressive search space reduction for human pose estimation. In: Proc. CVPR (2008)
41. Ioffe S., Forsyth D.: Finding people by sampling. In: Proc. ICCV, pp. 1092–1097 (1999)
42. Jiang H.: Human pose estimation using consistent max-covering. In: Proc. ICCV (2009)
43. Micilotta A., Ong E., Bowden R.: Detection and tracking of humans by probabilistic body part assembly. In: British Machine Vision Conference (2005)
44. Mikolajczyk K., Schmid D., Zisserman A.: Human detection based on a probabilistic assembly of robust part detectors. In: Proc. ECCV, pp. 69–82 (2004)
45. Mori G.: Guiding model search using segmentation. In: Proc. ICCV, pp. 1417–1423 (2005)
46. Ramanan D., Forsyth D.A., Zisserman A.: Strike a pose: tracking people by finding stylized poses. In: Proc. CVPR, pp. 271–278 (2005)
47. Ren X., Berg A.C., Malik J.: Recovering human body configurations using pairwise constraints between parts. In: Proc. ICCV, pp. 824–831 (2005)

48. Roberts, T.J., McKenna, S.J., Ricketts, I.W.: Human pose estimation using partial configurations and probabilistic regions. IJCV **73**(3), 285–306 (2007)
49. Ronfard R., Schmid C., Triggs B.: Learning to parse pictures of people. In: Proc. ECCV, pp. 700–714 (2002)
50. Daubney B., Gibson D., Campbell N.: Real-time pose estimation of articulated objects using low-level motion. In: Proc. CVPR (2008)
51. Ramannan D.: Learning to parse images of articulated bodies. In: Proceedings of neural information processing systems, pp. 1129–1136 (2006)
52. Yao B., Li F.F.: Modeling mutual context of object and human pose in human-object interaction activities. In: Proc. CVPR (2010)
53. Cham T.J., Rehg, J.M.: A multiple hypothesis approach to figure tracking. In: Proc. CVPR, pp. 239–245 (1999)
54. Difranco D.E., Cham T.J., Rehg J.M.: Recovery of 3-D figure motion from 2-D correspondences. In: Proc. CVPR (2001)
55. Ju S., Black M., Yacoob Y.: Cardboard people: a parameterized model of articulated motion. In: Proc. Automatic Face and Gesture Recognition, pp. 38–44 (1996)
56. Rehg J., Kanade T.: Model-based tracking of self occluding articulated objects. In: Proc. CVPR, pp. 612–617 (1995)
57. Lee M.W., Cohen I.: Proposal maps driven mcmc for estimating human body pose in static images. In: Proc. CVPR, pp. 334–341 (2004)
58. Felzenszwalb, P.F., Huttenlocher, D.P.: Pictorial structures for object recognition. Int. J. Computer Vis. **61**(1), 55–79 (2005)
59. Hua G., Wu Y.: Multi-scale visual tracking by sequential belief propagation. In: Proc. CVPR, pp. 826–833 (2004)
60. Isard M.: Pampas: Real-valued graphical models for computer vision. In: Proc. CVPR, pp. 613–620 (2003)
61. Sudderth E.B., Ihler A.T., Freeman W.T., Willsky A.S.: Nonparametric belief propagation. In: Proc. CVPR, pp. 605–612 (2003)
62. Sudderth E.B., Mandel M.I., Freeman W.T., Willsky A.S.: Visual hand tracking using nonparametric belief propagation. In: IEEE CVPR Workshop on Generative Model based Vision (2004)
63. Brubakerl, M., Fleet, D., Hertzmann, A.: Physics-based person tracking using the anthropomorphic walker. Int. J. Computer Vis. **87**(1), 140–155 (2010)
64. Fossati A., Fua P.: Linking pose and motion. In: Proc. ECCV (2008)
65. Gupta A., Chen T., Chen F., Kimber D., Davis L.S.: Context and observation driven latent variable model for human pose estimation. In: Proc. CVPR (2008)
66. Howe N.R., Leventon M.E., Freeman W.T.: Bayesian reconstruction of 3D human motion from single-camera video. In: NIPS (1999)
67. Sidenbladh H., Black M.J.: Learning image statistics for bayesian tracking. In: Proc. ICCV 2, pp. 709–716 (2001)
68. Sigal L., Bhatia S., Roth S., Black M.J., Isard M.: Tracking loose-limbed people. In: Proc. CVPR, pp. 421–428 (2004)
69. Urtasun, R., Fleet, D.J., Fua, P.: Monocular 3D tracking of the golf swing. In: Proc. CVPR **2**, 932–938 (2005)
70. Urtasun R., Fleet D.J., Hertzmann A., Fua P.: Priors for people tracking from small training sets. In: Proc. ICCV, pp. 403–410 (2005)
71. Taylor G., Sigal L., Fleet D., Hinton G.: Dynamical binary latent variable models for 3D human pose tracking. In: Proc. CVPR, (2010)
72. Gleicher M.: Retargeting motion to new characters. In: ACM SIGGRAPH, pp. 33–42 (1998)
73. Jones, M.J., Rehg, J.M.: Statistical color models with application to skin detection. IJCV **46**(1), 81–96 (2002)
74. Rother C., Kolmogorov V., Blake A.: Grabcut—interactive foreground extraction using iterated graph cuts. In: Proc. ACM SIGGRAPH, pp. 309–314 (2004)
75. Myers, C., Rabinier, L., Rosenberg, A.: Performance tradeoffs in dynamic time warping algorithms for isolated word recognition. IEEE Trans. Acoustic Speech Signal Process. **28**(6), 623–635 (1980)
76. Yang Y., Ramannan D.: Articulated pose estimation with flexible mixtures-of-parts. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (2011)

Author Biographies

Wee Kheng Leow obtained his B.Sc. and M.Sc. degrees in Computer Science from National University of Singapore in 1985 and 1989 respectively. He pursued Ph.D. study at The University of Texas at Austin and obtained his Ph.D. in Computer Science in 1994. His current research interests include computer vision, medical image analysis, and protein docking. He has published more than 80 technical papers in journals, conferences, and books. He has also been awarded two U.S. patents and has published another patent under PCT. He has served in the Program Committees and Organizing Committees of various conferences. He has collaborated widely with a large number of local and overseas institutions. His current local collaborators include I2R of A*STAR, Singapore General Hospital, National University Hospital, and National Skin Centre, and overseas collaborators include CNRS in France and National Taiwan University and National Taiwan University Hospital.



Ruixuan Wang received the B.Eng. and M.Eng. degrees from Xi'an Jiaotong University, Xi'an, China, in 1999 and 2002, respectively, and the Ph.D. degree from National University of Singapore, Singapore, in 2008. He is currently a Postdoctoral Research Assistant with the School of Computing, University of Dundee, Angus, U.K. His research interests include computer vision, image and video analysis, and machine learning.

Hon Wai Leong is an Associate Professor in the Department of Computer Science at the National University of Singapore. He received the B.Sc. (Hon) degree in Mathematics from the University of Malaya and the Ph.D. degree in Computer Science from the University of Illinois at Urbana-Champaign. His research interest includes the design of efficient algorithms for solving practical problems. The application areas include VLSI-CAD, transportation logistics, multimedia systems, and computational biology. In computational biology, his current research includes computational proteomics, fragment assembly, comparative genomics, and mining PPI networks. One of his passions is working with young people—giving outreach camps on creative problem solving skills and mentoring young student research projects. In Singapore, he founded the Singapore training program for the International Olympiad in Informatics (IOI). He is a member of ACM, IEEE, ISCB, and a Fellow of the Singapore Computer Society (http://www.comp.nus.edu.sg/~leonghw/).