Reconstruction of segmentally articulated structure in freeform movement with low density feature points

Baihua Li*, Qinggang Meng, Horst Holstein

Department of Computer Science, University of Wales, Aberystwyth, SY23 3DB, Wales, UK

Received 12 March 2003; received in revised form 14 January 2004; accepted 12 February 2004

Abstract

Though a large body of research has focused on tracking and identifying objects from the domain of colour or grey-scale images, there is a relative dearth in the literature on complex articulated/non-rigid motion reconstruction from a collection of low density feature points. In this paper, we propose a segment-based articulated matching algorithm to establish a crucial self-initialising identification in model-based point-feature tracking of articulated motion with near-rigid segments. We avoid common assumptions such as pose similarity or small motion with respect to the model, and assume no prior knowledge of a specific movement from which to restrict pose identification. Experimental results based on synthetic pose and real-world human motion capture data demonstrate the ability of the algorithm to perform the identification task.

Keywords: Articulated point matching; Non-rigid pose estimation; Affine transformation; Motion tracking and object recognition; Motion capture

1. Introduction

Reconstruction of articulated/non-rigid objects has lately been an important issue in computer vision, motivated by a number of applications such as human-machine interaction, biomedical studies, the entertainment industry, security systems, and more recently, robot monitoring and control. A large body of research is dedicated to this task in the usual domain of projective colour or grey-scale images [1–3]. Most existing algorithms, for instance in the field of ‘looking at people’, have been designed to deal with problems such as human body detection, motion reconstruction, pose estimation and motion recognition, using richer information from images [4–7]. In image-based systems, feature points might be used as an intermediate correspondence to assist the processing towards an end-product motion and structure representation.

There are many cases, such as in machine vision, graphics, astronautics, computational biology and computational chemistry [8–10], in which a complex 2D or 3D articulated/non-rigid structure and motion are represented only by point features, obtained by stereo vision or other sensor systems. In this context, the spatio-temporal information is reduced simply to unidentified (unlabelled) points moving over time. There is a relative dearth of literature on articulated motion reconstruction from only sparse point features [11]. Feature-point tracking and identification are required to determine underlying subject structures in such systems. Based on assumptions such as small or smooth inter-frame motion, or high-level knowledge related to a specific motion [12,13], point-feature tracking has been investigated extensively. However, the identities of the subject feature points are not available from the inter-frame tracking alone. Feature point identification, to know which point in an observed data frame corresponds to which point in its model and thence recovery of structure, is inherently difficult and still remains an open problem, especially at the start or recommencement of tracking in the real-world. Currently, most tracking approaches simplify the problem to incremental pose estimation, relying on manual model fitting at the start of tracking, or an assumption of initial pose similarity to the model, or pre-knowledge of a specific motion from which to infer an initial pose estimation.
We concentrate on the identification task to address the crucial problem of self-initialisation in sparse feature-point tracking of articulated motion with near-rigid segments. Therefore, our algorithm assumes availability of non-projective feature point data for such motion obtained by appropriate sensors, such as the 3D data obtained in our experiments via a marker-based optical motion capture (MoCap) system. The articulated subject to be monitored is a priori known, suggesting a model-based approach. We formulate this problem as one-to-one point pattern matching (PPM) between the model data and pose data taken from its freeform movements.

2. Research background

PPM is a fundamental, commonly encountered, yet still open problem covering both theoretical and practical issues. It is often advantageous to represent sets of image features (extrinsic or intrinsic [3]) by their coordinates. Although object recognition relies on PPM, motion estimation and image registration, many of the techniques have focused on rigid, affine or projective point correspondence, or pose estimation using dense point-sets on curves or surfaces [14–16]. These methods are based on geometric invariance or constraint satisfaction embedded in affine transformations, yielding approximate matches among objects and models [17].

Compared with rigid and affine motion, the articulated/non-rigid motion expresses a mathematically more complex relation. To make the problem tractable, most methods are application dependent along with certain specified global/local restrictions relating to movements, objects and data properties. A global formulation may be approximated via piecewise approximation, elastic/deformable models, or weighted-graph matching [10]. The piecewise methodology is probably the most popular approach in non-rigid applications for piecewise segmented data, since its computational complexity is relatively low and its implementation relatively easy [18–20]. Piecewise approximation performs computation, evaluation and optimisation for each locality and successively propagates to different neighbourhoods. High order piecewise polynomials, such as splines, or piecewise affine transformations are frequently used for modelling and decomposing local geometric distortion. Feldmar and Ayache [21] have argued the possibility of enhancing the piecewise approach by overriding global information.

Marker-based optical MoCap systems [22,23], as currently used in clinical gait analysis, sports studies, computer game and animation industries, virtual/augmented reality, are examples of 3D point-feature trackers. During MoCap, markers are attached on a subject as feature points. Such systems provide an impressive facility for accurately recording the subject’s movement in ‘real-time’, represented by sequences of 3D feature-point coordinates, but reconstruction of the subject’s motion requires an additional step - identification. Some identification algorithms may work well for a simple marker patterns under rigid motion. For complex motion exemplified by human motion, methods using coloured markers, active markers or a set of unique spatial combinations of point-patterns [24], may be employed to ‘code’ the identification information. These methods, requiring specially designed markers or point-pattern distributions, avoid the hard problem of determining articulated/non-rigid motion directly from the generic structure feature-points, but at the possible cost of application restriction. The generic PPM problem in articulated motion is exemplified by a state-of-the-art MoCap system such as the Vicon-512 System [22], employing passive infrared reflective markers. It uses a model-based approach to label the captured motion data. The model itself is defined from one captured frame of a design pose of the subject, with embedded articulated structure information. The model points are usually labelled manually. The model serves as an individual pseudo-skeleton for feature-point identification in dynamic MoCap trials of the subject. Auto-identification may fail for complex and noisy data. Captured data normally need time-consuming manual post-processing and correction before they can be used for actual applications [25].

Fitting the individual model to its massive motion data is a routine but hard task for a number of reasons. A practical tracker should have the following facilities: self-starting without any tracking initialisation assumption; capable of recovering the articulated motion of non-rigid segments rather than restrict to only rigid movements; tolerant to some degree of corrupted data in real-world situations with missing (occluded) and extra (system noise) data; automatically tracking through free-form movements rather than pre-known specific motion or starting from a design pose; computational efficiency. Existing PPM and related tracking algorithms will find difficulty in meeting all these criteria. This is the motivation for seeking a novel algorithm for segment-based articulated PPM, drawing on the rich experience of existing PPM techniques.

In this paper, we present a segment-based articulated model matching algorithm to establish the precise one-to-one match between the model point-set and its observed pose data. In our algorithm, a segment-based ‘stick-figure’ model of an observed subject is pre-defined off-line (Section 3.1). The observed data is taken from the freeform motion of the articulated subject, represented by low-density feature points and the same dimensionality as its model. We address this problem in real-world situations with occlusion, noisy data and limited distortion in each segment. We do not make assumptions regarding model pose similarity or restrictions to a specific motion. This work contributes to the crucial task of self-initial identification in articulated tracking, noting that after the initial identification, identities of feature points can be easily maintained from inter-frame tracking.
Though subsequent occlusion may cause some broken trajectories and loss of feature point identities, the complexity of re-identifying the occluded points is much reduced compared to the initial stage when no prior identification exists.

In the following, Section 3 describes the model and formulates the objective in a best-fit sense. Section 4 presents the algorithm in detail. In Section 5, we provide experimental results and discussions on both synthetic and real human MoCap data. Section 6 concludes the work.

3. Framework of the algorithm

3.1. ‘Stick-figure’ model for an articulated structure with near-rigid segments

The articulated structure we are considering undergoes segment-based jointed motion. To keep the model-based identification algorithm applicable to a wide range of subjects with articulation, we allow each segment during movement to be rigid or near-rigid within a distortion tolerance ratio $e_i$, and to undergo a high-dimensionally independent transformation constrained only by inter-segmental joints. To keep the algorithm general, we do not impose motion constraints such as feasible biological poses for a specific subject type.

When we use sparse points as features to represent such articulated motion, feature points (either extrinsic or intrinsic) commonly locate some key positions, such as joints between articulated segments [13]. The feature points on a segment must be sufficient in number and distribution to describe the required orientation and required detailed segmental structure. For the model, we generate off-line the geometrical structure of an articulated subject as a ‘stick-figure’, using one complete frame of identified pose coordinate data and linking pairs of intra-segmental feature points into groups corresponding to the underlying articulated ‘anatomy’. Articulation is indicated through joint-point commonality between two segments, suggesting a segment-based hierarchy, as shown in Figs. 1 and 4 (left).

More formally, we define the model point-set $\mathcal{P}$ to consist of a set of $S$ segments: $\mathcal{P} = \{\mathcal{P}_s\}_{s=1,\ldots,S}$. Each segment $\mathcal{P}_s = \{e_i, (p_{si}, L_{si})\}_{i=1,\ldots,M_s}$ has a distortion tolerance ratio $e_i$, and $M_s$ model points described by their coordinates $p_{si} \in \Omega$ and identification labels $L_{si}$, where $\Omega$ states the dimensionality of the problem, being one of either $\mathbb{R}^2$ or $\mathbb{R}^3$.

3.2. Integration of hierarchical search with segment-based best-fit objective

We denote the observed data set $\mathcal{Y} = \{q_j| q_j \in \Omega, j = 1,\ldots,N\}$ of $N$ unidentified points in a randomly sampled frame of the modelled subject during its freeform movement with real-world noise.

To fit the pre-defined segment-based ‘stick-figure’ model to its related pose data, matching at segment level is feasible and preferable to inefficient brute global point-to-point searching. Therefore, the proposed segment-based articulated model-matching algorithm includes two steps: firstly, to establish local segmental one-to-one correspondences $\{(p_{si}, q_{sj})\}$ by looking for the best transformation that interprets near-rigid segmental movement (Section 4.1); and secondly, to apply a segment-based depth-first search constrained by the model’s hierarchical structure tree (Section 4.2).

We define the best segmental transformation under associated correspondences by two criteria: matching quality $\tilde{e}$ and matching size $\Theta$. For matching quality, we require the average matching error $\tilde{e}$ between segment $\mathcal{P}_s\{p_{si}\}$ and its assumed match $\mathcal{Y}_s\{q_{sj}\}$, to be less than a specified relative segmental distortion threshold $e_s$,

$$
\tilde{e}_s = \frac{\sum_{j=1}^{M_s} r_{sj} e_{sj}}{\sum_{j=1}^{M_s} r_{sj}} < e_s, \tag{1}
$$

where $e_{sj} = d(q_{sj}, R p_{si} + T_i) / l_i$ is the relative matching error in the assumed match $(p_{si}, q_{sj})$ under the affine transformation $[R_s, T_i]$, with $d(\ldots)$ denoting Euclidean distance and $l_i = \text{avg}_{j}[d(p_{si}, q_{sj})]$ denoting the average intra-point distance in model segment $\mathcal{P}_s$; and match indicator $r_{sj}$ is a value of 1 or 0 to indicate the presence or absence of an assumed match $(p_{si}, q_{sj})$ between model point $p_{si}$ and data point $q_{sj}$. The affine transformation $[R_s, T_i]$ ($R_i$ for rotation, $T_i$ for translation) is computed from the least-squares objective function,

$$
[R_s, T_i] = \arg \min_{R,T} \frac{\sum_{j=1}^{M_s} r_{sj} d^2(q_{sj}, R p_{si} + T_i)}{\sum_{j=1}^{M_s} r_{sj}} \tag{2}
$$

For matching size, we require that the fraction $\Theta_j$ of non-null matches $\mathcal{Y}_s$ satisfies the condition, with specified $\beta \in (0, 1)$,

$$
\Theta_j = \frac{\sum_{i=1}^{M_s} r_{sj}}{M_s} > \beta. \tag{3}
$$

When both conditions are satisfied, we regard $\mathcal{Y}_s\{q_{sj}\}$ as the best segmental match of $\mathcal{P}_s$, and denote the correspondence by $\{(\mathcal{P}_s\{p_{si}\}, \mathcal{Y}_s\{q_{sj}\})\}$. 

4. The segment-based articulated matching algorithm

4.1. Segmental matching by iteration

Matching is commonly considered as an efficient search and verification procedure based on geometric invariance embedded in rigid or affine transformations [17]. In the case of articulated near-rigid motion, the geometric invariance is approximate and local to segments. Moreover, segmental matching may have to process a large number of outliers arising from the data of other as yet unmatched segments. It is practical initially to reduce the search space by excluding some outliers without compromising the correct solution to the correspondence problem, before actually invoking a heuristic search and verification method to find the correspondence.

We notice that using complex geometric constraints to find such a subspace could lead to a large interpretation search tree that is not always computationally efficient [17]. We adopt a relaxed segmental intra-distance invariance criterion to restrict the search space to optimised candidate-tables (CTs) for segmental matching (refer to Section 4.1.1) as the first stage. Intra-distance, measured from an assumed 'pivot' match as reference data point, allows exploration of all data distances using only this one match assumption. Had we used complex geometric constraints based on more assumed reference points, angle for example, there would be combinatorial combinations for interpretation. The cost of finding and ranking the CTs does not outweigh the resulting saving in search. Moreover, initial distance-bounding can be efficiently applied to reduce outliers for segment-based data distribution.

In the second stage, to search, verify and refine segmental matches, an iterative heuristic search procedure is applied to the optimised subspace CTs (Sections 4.1.2–4.1.4). The segmental motion estimate is calculated by least-squares fitting under the assumed matching pairs. The fit must satisfy a predefined match tolerance to confirm the match assumption, and must be able to refine a probable partial match to whole match to verify the match against the geometric constraint of the model segment.

4.1.1. Candidate-table generation and optimisation

To identify a segment $\mathcal{P}_s$ in an articulated structure, as in Fig. 1\(^1\), we list data points in a table as candidates for assumed matches of model points as follows:

\[
\begin{align*}
\text{Step 1.} & \quad \text{Create an ordered pivotal sequence for segment } \mathcal{P}_s : \\
& \text{If no point has been identified in segment } \mathcal{P}_s, \text{ arbitrarily choose a reference point } p_{s,1} \in \mathcal{P}_s, \text{ named 'pivot', and order the remaining points } p_{s,j} \text{ in the segment by non-decreasing distance } d(p_{s,i}, p_{s,j}) \text{ from the pivot, to define an ordered pivotal sequence for the segment. The case of an already identified point is considered at the end of this sub-section.}
\end{align*}
\]

\[
\text{Step 2.} \quad \text{Find candidates and generate CTs in a pivot centred subspace:}
\]

In the unidentified observed data $\mathcal{J}_R$, choose an assumed pivot match $q_i$. As a pre-processing step in CTs generation, restrict the search space for the match of segment $\mathcal{P}_s$ by excluding large outliers in $\mathcal{J}_R$. Thus, we apply a $q_i$-centred distance-bound to the unmatched point-set $q_i \in \mathcal{J}_R$ to find a bounding subset $\mathcal{J}_M$

\[
\mathcal{J}_M = \{ q_i | d(q_i, q_j) < (1 + 2\varepsilon)M_{s,i} \},
\]

where $M_{s,i} = \max_i \{ d(p_{s,i}, p_{s,j}) \}$ is the pivot-centred distance bound for segment $\mathcal{P}_s$. Then, find match candidates $q_j$ for each $p_{s,j}$, using a relaxed tolerance criterion,

\[
\frac{|d(p_{s,i}, p_{s,j}) - d(q_i, q_j)|}{d(p_{s,i}, p_{s,j})} < 2\varepsilon; \quad q_j \in \mathcal{J}_M.
\]

and list them in a table column indicating match candidates for element $p_{s,j}$. The procedure is repeated for every element of the pivotal sequence, giving rise to an ordered sequence of columns that define the CT for the pivot choice $p_{s,i}$ and its assumed match $q_i$.

An example CT for an assumed match $q_1$ of pivot $p_{s,1}$ is shown in Table 1 for identifying segment $\mathcal{P}_s$ in Fig. 1. The underlined points indicate correct matches in the CT, to be discovered by the algorithm.

\[
\text{Step 3.} \quad \text{CTs optimisation:}
\]

By Step 2, taking each unmatched point in $\mathcal{J}_M$ in turn as an assumed pivot match of $p_{s,i}$, we generate a set of CTs for that pivot choice. If the match of pivot $p_{s,i}$ is not lost, the CT constructed with the correct pivot match should contain the correct interpretation of segment $\mathcal{P}_s$. Heuristically, this CT has more candidates than other CTs.

To economise the search for a likely correct solution: (1) discard CTs with low numbers of candidates (CT culling);

![Fig. 1. Segment $\mathcal{P}_s$ identification. Model (left), observed data (middle), result (right).](image)

\(^1\) The enumeration subscripts of corresponding model and data points are chosen to be the same in Fig. 1, for ease of illustration.

### Table 1

<table>
<thead>
<tr>
<th>Pivotal sequence: $p_{s,1}$</th>
<th>$p_{s,2}$</th>
<th>$p_{s,3}$</th>
<th>$p_{s,4}$</th>
<th>$p_{s,6}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Increasing $d(p_{s,1}, p_{s,j})$:</td>
<td>198</td>
<td>220</td>
<td>265</td>
<td>338</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CT($q_1$):</th>
<th>$q_1$</th>
<th>$q_2$</th>
<th>$q_3$</th>
<th>$q_4$</th>
<th>$q_6$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$q_5$</td>
<td>$q_7$</td>
<td>$q_9$</td>
<td>$q_9$</td>
<td>$q_6$</td>
</tr>
<tr>
<td></td>
<td>$q_5$</td>
<td>$q_{10}$</td>
<td>$q_6$</td>
<td>$q_{11}$</td>
<td>$q_{10}$</td>
</tr>
</tbody>
</table>
If a join point in the segment \( \mathcal{P}_i \) has already been identified during its parent-segment identification, reasonably this point is chosen as the pivot in Step 1. Therefore, in Step 2 only one CT need be generated for the segment, based on this known pivot match. This implies a striking reduction of the search space.

Using the intra-point distance similarity criterion, Eq. (5), the CTs set up the correspondence assumption between a model segment and the data. The process works even in the presence of a large number of outliers from the data of unmatched articulated segments, and regardless of the pose. This renders unnecessary the assumption of small motion or pose similarity, used for example in the Iterative Closest Point (ICP) algorithm [16]. CTs optimisation reduces the search space and improves efficiency of the iterative search, through CT culling, ranking and candidate ordering.

### 4.1.2. CTs-based iterative segmental matching

The CTs encompass a set of matching assumptions. To detect a one-to-one segmental match from the prioritised CTs, we apply an iterative matching procedure, as shown in Fig. 2, with two functions:

1. **Iterative motion estimation and verification.** Determining a transformation \([\mathbf{R}_n, \mathbf{T}_n]\) in a least-squares manner by the Singular Value Decomposition (SVD) algorithm [26], when at least three assumed matching pairs are available in a CT. This transformation should map model points into a rough alignment, with their assumed matches satisfying a pre-defined segmental-distortion bound as in Eq. (1). Otherwise, there must be pseudo pairs in this assumption. Accordingly, update matching pairs from the CT, as in Section 4.1.3.

2. **Refinement.** The verified matching pairs may only give a partial match of the segment. To achieve the matching size criterion in Eq. (3), a guided exploration of the unidentified data by the obtained \([\mathbf{R}_n, \mathbf{T}_n]\) searches for additional matches, as in Section 4.1.4.

Motion estimation and refinement alternately update the assumed matches in iterations, on successive CTs if necessary, until the best match \((\mathcal{P}_i, \mathcal{J}_i, \mathcal{Q}_i)\) is found if it exists, satisfying criteria both of matching quality in Eq. (1) and matching size in Eq. (3).

#### 4.1.3. Update of an assumed correspondence and rejection of pseudo matches

To update an assumed correspondence from a CT in Fig. 2, two methods are used. When a segment has more than a few points, we found for an assumed correspondence with some spurious pairs that the average matching error \(\bar{e}_a\) generally exceeds greatly the segmental distortion tolerance \(e_s\) in Eq. (1), and exaggerates matching error \(e_{a,i}\) at wrong matches. Based on this cue, we remove the worst match by replacing it with its next candidate in the CT, if it exists; otherwise we omit its match from the currently assumed correspondence.

By this method of ‘coarse’ updating, the algorithm can quickly reject pseudo matches and explore and eliminate unfavourable CTs, to converge to a segmental match in the correct CT for a large segment. This is especially significant when there is a large number of CTs to be investigated, such as when matching the first segment, or a subsequent segment without identified or recovered virtual join points (Section 4.2).

The reliability of this heuristic updating is subject to the number of matching pairs involved and data quality. The method infers pseudo matches indicated by transformation errors on the basis of the least-squares technique. The larger the distortion encountered in a segment, the greater the number of match pairs needed for accurate motion estimation, to avoid masking out correct matches or accepting false matches. In human motion with average segmental distortion about 5–15%, experimental results showed an adequate threshold of matching pairs used in this heuristic updating to be 6.
Table 2
Iterative updating of assumed matches from the CT in Table 1

<table>
<thead>
<tr>
<th>Iteration (1) initial matches:</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{M}_M = {q_1, q_2, q_3, q_4, q_5}$</td>
</tr>
<tr>
<td>matching error: $e_r = [0.57 0.68 0.38 1.98 0.46 0.52]$</td>
</tr>
<tr>
<td>$\tilde{e}_r = 0.765 &gt; e_r$, maximum $e_r$, at $q_4$.</td>
</tr>
<tr>
<td>Iteration (2) updated matches:</td>
</tr>
<tr>
<td>$\mathcal{M}_M = {q_1, q_2, q_3, q_4, q_5}$</td>
</tr>
<tr>
<td>matching error: $e_r = [0.43 0.38 0.25 0.08 0.15 0.87]$</td>
</tr>
<tr>
<td>$\tilde{e}_r = 0.36 &gt; e_r$, maximum $e_r$, at $q_4$.</td>
</tr>
<tr>
<td>Iteration (3) updated matches:</td>
</tr>
<tr>
<td>$\mathcal{M}_M = {q_1, q_2, q_3, q_4, q_5}$</td>
</tr>
<tr>
<td>matching error: $e_r = [0.04 0.05 0.05 0.03 0.07 0.04]$</td>
</tr>
<tr>
<td>$\tilde{e}_r = 0.047 &lt; e_r$.</td>
</tr>
<tr>
<td>End of updating $\mathcal{M}_M$.</td>
</tr>
</tbody>
</table>

The iterative updating of $\mathcal{M}_M$ by the first method is illustrated in Table 2 for the CT in Table 1, with the distortion tolerance ratio $e_r$ set to 0.1.

For a segment of only a few points and/or of poor rigidity, the matching errors can distribute more evenly so as to hide outliers, and so the cue used in the 'coarse' updating becomes unreliable. Correct matches might be ruled out, with rapid depletion of match choices. In this case we recur to a second method: pivot-nearest-furthest sequential search. This takes advantage of the optimal ordering technique employed for CT generation (refer to Section 4.1.1).

In this method, each iteration uses only three points $\{q_1, q_{\text{near}}, q_{\text{fur}}\}$ whose columns define the assumed correspondence in the model pivot sequence, to provide an assumed partial match. The search routine is shown in Fig. 3, where $CT_{m,n}$ indicates the candidate element on row $m$ and column $n$. The rationale for choosing the pivot’s nearest and furthest neighbours is that such points in a segment have less ambiguity then others with intermediate distances [14, 15, 17].

* $N_r$ is the number of rows of the CT
* $N_c$ is the number of columns of the CT

\[
\begin{align*}
q_r & \leftarrow CT_{1,1}; \quad \text{* assumed pivot match} \\
\text{for } c_1 = 2 & \text{ to } N_c - 1 \\
\text{for } r_1 = 1 & \text{ to } N_r \text{ while notNull}(CT_{r_1,c_1}) \\
q_{\text{near}} & \leftarrow CT_{r_1,c_1}; \quad \text{* second point} \\
\text{for } r_2 = 1 & \text{ to } N_r \\
\text{for } c_2 = N_c & \text{ to } c_1 + 1 \\
\text{if notNull}(CT_{r_2,c_2}) & \\
q_{\text{far}} & \leftarrow CT_{r_2,c_2}; \quad \text{* third point}
\end{align*}
\]

Fig. 3. Sequential search for pivot-nearest-furthest candidates (* for comments).

4.1.4. Refine a partial match to a whole match of the segment

The above search procedure may have dropped correct matches during the iterations in the first method, or have omitted them in the second method, or may have excluded them on grounds of limiting the CT search space by Eq. (5). To improve a partial match to a whole segmental match satisfying the matching size criterion Eq. (3), we apply the obtained motion estimate $\{R_s, T_s\}$ from the partial match to all the points in model segment $\mathcal{P}_s: \mathbf{p}_{ij} = R_s \mathbf{p}_{ij} + T_s$, and reassign their closest-neighbour matches $\{\mathbf{p}_{ij}, q_{ij}\}$ in $\mathcal{M}_M$, satisfying

\[
d(\mathbf{p}_{ij}, q_{ij}) = \min_{q \in \mathcal{M}_M} d(\mathbf{p}_{ij}, q) < 2l_i e_i. \quad (6)
\]

If no such closest neighbour is found, we say the match point of $\mathbf{p}_{ij}$ is lost, and set the match indicator $r_{ij}$ to 0.

4.2. Hierarchical segment-based depth-first tree search

If we restrict matching only to the level of segment-based optimisation, uncertainties can arise when a segment contains only a few points or has strong similarity with other segments. Matching articulated segments requires consistency with model parent-child segment connections, presentable in a hierarchical tree, encapsulating the underlying geometric articulation.

We assume in a segmentally articulated structure that one of the segments contains more points and has more segments linked to it than most other segments. We treat such a segment as root, and seek to identify the root segment at the first stage of a hierarchical search. After the root has been located, searching proceeds depth-first to child segments along hierarchical chains, taking advantage of available joints or virtual joints, which have been located during parent-child identification. Virtual joints are SVD-recoverable missing joints in parent segments. This linkage through join points considerably increases the reliability and efficiency of child-segment identification. When a parent has several children, searching priorities to the child with the most model points, because identification is more reliable and the subsequent search space is reduced more rapidly.

When a segment includes only few points, such as point-pair segments or segments with several missing points, the SVD-based algorithm (requiring at least three matching pairs) does not work. Segmental identification may be highly uncertain with such reduction of information. In this case, we need to confirm such a segment in the hierarchical chain depending on whether its children or even grand-children can be found.

Failure to identify a descendant may imply a wrong parent-child identification. In this case, the algorithm attempts a backward correction to the parent segment. In the worst situation when a search chain in the hierarchy is broken by a failed segment identification, hierarchical
searching will tend to identify other segments on other chains, leaving any remaining child-segments on broken chains the last to be resolved.

Our algorithm makes use of a general jointed articulated segmental model. In some applications restricted to a specific subject type, such as human, additional information, such as feasible pose or motion constraint, is available, and could be imposed on our identification framework to reduce identification ambiguities.

5. Experiments

We have implemented the proposed algorithm in Matlab and applied it to both synthetic pose data and real-world human pose data. In our experiments, all model data and motion data are acquired from a marker-based optical MoCap Vicon-512 system. It includes 7 high-resolution calibrated CCD cameras. The system can reconstruct the 3D-coordinates of each infrared-reflective marker attached on a subject, if the marker is located in at least two camera views. The measurement accuracy of this system is to a level of a few millimetres in a control volume spanning metres in linear extent.

5.1. Synthetic pose data

To evaluate the robustness and efficiency of the algorithm, we used human data to define two models of 13 segments, one having 31 points and the other being a denser case of 50 points. Both data sets were obtained from the same subject in similar poses, for comparability.

5.1.1. Identification of distortion-free arbitrary poses in 2D and 3D

In the first series of experiments, we studied the ability of the proposed algorithm to identify arbitrary distortion-free synthetic poses. A number of articulated models determined from various subjects and point pattern distributions were

(a) 2D sparse example: 13 segments and 31 points.

(b) 3D dense example: 13 segments and 50 points

Fig. 4. Arbitrary pose identification with synthetic data: model (left), observed data (middle), identified pose data (right).
investigated. To generate synthetic pose data, we applied a common translation to all model points and then applied a random rotation to each segment around its joint. We show example results for synthetic data of the two human models in Fig. 4, in which identified feature points are shown by intra-segment stick links. The algorithm is adaptable to 2D articulated structures and motion—a degenerate case of 3D. An example is shown in Fig. 4a. We obtained the 2D-model data by projecting 3D-model data on to a constant third dimension, and applied 2D articulated transformations embedded in this plane. Tests with thousands of such synthetic poses in both 2D and 3D, corrupted with about 40% spurious extra data in both sparse and denser point-sets showed that correct identification occurred in about 98% of the synthetic poses. We note that this high identification rate was achieved without any pose similarity between the model and its distortion-free data.

5.1.2. Identification of distorted pose data

The second series of experiments demonstrates the effect on identification of added segmental distortion, for a sparse and a denser point-sets in 3D. For each point in a model segment \( P_s \), we added zero-mean Gaussian noise \( N(0, 0.5\delta_l)\sqrt{6} \), with a standard deviation scaled by distortion level, \( \psi \), and the average segmental length, \( l_s \), to its Cartesian coordinates \( x, y \) and \( z \), respectively. The average identification rate (fraction of correctly identified data) versus increasing distortion level of both model sets are given in Fig. 5. Each experimental point is based on the average of 500 random configurations on a given distortion level, for 11 distortion levels \( \psi \) in the range 0–30% of \( 0.5l_s \). We observe that up to a 20% distortion level, both sparse and denser sets achieve similar identification rates. However, with further increasing distortion, there is more potential for confusion among neighbouring data points in the denser point set, leading to greater loss of identification.

5.1.3. Corrupted pose data with missing/extra data and distortion

The third series of experiments illustrates the performance of the algorithm in identifying sparse (31 points) and denser (50 points) data sets under the corruption of missing or extra data. To obtain such data, we first transformed model data to generate an arbitrary pose as in Section 5.1.1; secondly applied 6% distortion on each data point as in

Section 5.1.2; and lastly, randomly removed or added some points. The extra points were added in a volume encompassing the observed data. Identification rates for such data are shown in Fig. 6. The trend curves are based on average results for 500 randomly generated missing/extra data sets at each indicated experimental point, for a chosen pose. From Fig. 6, the observed mean identification rate (\( \times \)) decreases with increasing missing data from the zero reference line (indicating complete data). On the right side of the zero-line, the identification rate reduces smoothly with increasing added extra data. As may be expected, the gradients on the left and right sides of the zero-line are quite different: the algorithms robustly rejects large numbers of outliers (\( \sim 100\%) \), but cannot survive the inherent difficulty of missing data.

Comparing the cases of 31 and 50 point-sets, the denser point-set is capable of tolerating more missing data and maintains a better identification rate for a given missing data percentage, as is to be expected from the insufficient data content of the seriously depleted sparser point set. On the other hand, extra data show more identification loss for the denser set, demonstrating the increased inherent potential for confusion.

5.1.4. Efficiency

Computational complexity of the algorithm depends on the definition of the articulated model (the number of segments, the number of points and their distribution in each segment), but critically also on the experimental conditions: the level of encountered segmental distortion, pose complexity, and missing/extra data. These factors may result in
a varied search and identification cost. We can attempt to measure the complexity empirically via the total number of SVD invocations, as \textit{match count}, during articulated hierarchical searching.

We undertook complexity trials relating to the two experiments of Sections 5.1.2 and 5.1.3 above. We monitored match counts versus distortion of the sparser and the denser point-sets as shown in Fig. 7. Match counts are seen to grow linearly in log-scaling with distortion. The denser point set has the same number of segments, but will lead to more populated CTs. This is likely to require greater numbers of match attempts, but the overall complexity is seen not to grow as some high power of the increased number of model points. We believe this is because a whole segment match could take advantage of the guidance found from a partial match (Section 4.1.4).

Match counts versus missing/extra data are presented in Fig. 6. For both sparse and dense cases, we observe match counts grow steadily with increasing extra data in an approximate log-linear manner, however, match counts are bumpy for missing data. On the one hand, missing data reduce the number of points to be identified, but on the other, incomplete data might cause more identification difficulties, especially for missing join points and segments with small numbers of points.

We also monitored the CT optimisation using the experiment of Section 5.1.3 for the dense case. The results are reported in Fig. 8. The general trend with increasing missing or extra data is towards higher numbers of CTs generated ($\Delta$). The number of CTs remaining to be inspected after CT culling is remarkably reduced, reflected by the ratio of CTs optimised versus generated ($\bigcirc$). The low ratio of CTs inspected versus generated ($\times$) further demonstrates the efficiency of CTs ranking and candidate ordering in the CTs optimisation. We may conclude that the heuristic CT optimisation of Section 4.1.1 has proved highly effective in this synthetic case study.

5.2. Identification of human MoCap data in real-world

Human motion is a typical segment-based articulated motion, with segmental near-rigidity caused by deformable skin, muscle and loose clothing, etc. In our experiments, extrinsic markers, as feature points, were attached to human subjects at external sites chosen to limit distortion and reflect structure. The subject motion is recorded in a multi-camera stereo system to mitigate occlusion. To provide experimental values of the segmental distortion parameter $e_i$, we analysed a number of dynamic trials sampling various subjects and their movements. We found the segmental distortion differs with different body parts and motion intensity. Torso and thigh segments may result in large distortion with $e_i < 0.15$. We found a value of $e_i \approx 0.05$ to be adequate for the head, and an average $e_i \approx 0.1$ for other body parts.

For model generation in our experiments, for each subject, we took one complete frame of feature point data in a design pose, and manually identified the data in a 3D-interactive display. We used a predefined marker protocol to group the points into segments, giving an individual subject model consistent with the underlying articulated anatomy. Having attached markers to the subject under investigation and defined its model, we proceeded with the subject’s MoCap trials. These furnished the observed pose data, from which we randomly chose frames of the subject’s freeform movement sequences for analysis.

We investigated a number of subjects, movements and feature-point distributions. Illustrative results from
representative full-body marker protocols are shown in Fig. 9. The captured data may be subject to missing/extra data and segmental distortion. From the results, we observe that the proposed algorithm is capable of identifying an articulated pose and can deal with noisy data to some degree. Even when some key points, such as join points, have been lost, the algorithm can still carry on the child-segment identification successfully, by virtue of recovered virtual join points.

5.3. Parameter choice and execution time

The matching algorithm makes use of parameters $\beta$ and $\epsilon_s$, which control its performance in an intuitive way. Generally, for articulated subjects with rigid segments (e.g. robot manipulators), distortion tolerance $\epsilon_s$ in Eq. (1) can be small to raise the ability of rejecting outliers, hence the matching size requirement $\beta$ in Eq. (3) can be low to allow for more missing data. For subjects with deformable segments (humans), we raise the distortion ratio $\epsilon_s$ in order to relax the matching quality criterion, but at the cost of generating bigger CTs (Eq. (5)) and increased search burden. That may still result in some false matches. In this case, we raise the matching size parameter $\beta$ to demand improved evidence for a correct segmental match with, however, decreased ability to handle missing data.

The execution time depends not only on the number of points to be identified and their distribution in the segments, but also on the level of distortion, pose complexity and quality of data, as demonstrated by the match counts in Figs. 6 and 7. In particular, Fig. 7 shows an exponential time dependence on distortion level. When we applied our algorithm on human MoCap data in the real-world, we chose $\epsilon_s : 0.05 \sim 0.15$ and $\beta : 0.80 \sim 0.90$ based on the data quality with extremes ~10% missing/extra data per frame. Execution time in Matlab code on a 866 MHz Compaq with 256 MB of RAM, for identifying a single frame of pose data of full-body human motion data with 30–50 points, was in the range of 0.5 ~ 5 seconds.

6. Conclusions

We have proposed a new algorithm of segment-based articulated point matching for reconstructing a pose with low density feature points. It deals with the problem of self-initialisation and pose reconstruction in feature-point motion tracking. Self-initialisation is ultimately crucial for complete motion sequence recovery and determining tracking robustness. This aspect of the identification problem has received little attention in the literature.

Our algorithm employs several heuristic techniques to reduce the search space and achieve some robustness to data noise. The algorithm addresses identification of articulated motion in near-rigid segments using the articulated structure to organise a hierarchical segment-by-segment search framework, constrained by inter-segmental join points. This linkage significantly constrains the articulated child-segment sub-search, and improves the reliability of consistent segment identification. At the level of segment identification, we use powerful intra-point distance similarity heuristics to reduce and prioritise the search space. A relaxed motion estimation is used to search and confirm the identification of a segment, effectively rejecting outliers in the presence of distortion, missing and extra data.

Our algorithm addresses model-based matching in real-world situations with missing/extra data and limited distortion in segments. We do not make the common simplifying assumptions for model-fitting, such as pose similarity or small motion, nor do we assume any prior knowledge specific to the motion. The algorithm remains a candidate for on-line application in model-based point-feature trackers. For future work, to reduce the uncertainty inherent in the matching problem for single frame data, it suggests utilising start-up model matching enhanced by inter-frame tracking to lead to a robust self-initialising tracker.
Acknowledgements

All model data and motion data used in this paper were obtained by a marker-based optical MoCap system—Vicon 512, manufactured by Vicon Motion Systems Ltd, installed at the Department of Computer Science, UWA. We wish to thank Prof. Marcos Rodrigues at Sheffield Hallam University for his constructive and valuable suggestions and discussions.

References