Tracking and Motion Estimation of the Articulated Object: a Hierarchical Kalman Filter Approach

Real-time motion capture plays a very important role in various applications, such as 3D interface for virtual reality systems, digital puppetry, and real-time character animation. In this paper we challenge the problem of estimating and recognizing the motion of articulated objects using the optical motion capture technique. In addition, we present an effective method to control the articulated human figure in realtime.

The heart of this problem is the estimation of 3D motion and posture of an articulated, volumetric object using feature points from a sequence of multiple perspective views. Under some moderate assumptions such as smooth motion and known initial posture, we develop a model-based technique for the recovery of the 3D location and motion of a rigid object using a variation of Kalman filter. The posture of the 3D volumetric model is updated by the 2D image flow of the feature points for all views. Two novel concepts – the hierarchical Kalman filter (HKF) and the adaptive hierarchical structure (AHS) incorporating the kinematic properties of the articulated object – are proposed to extend our formulation from the rigid object to the articulated one. Our formulation also allows us to avoid two classic problems in 3D tracking: the multi-view correspondence problem, and the occlusion problem. By adding more cameras and placing them appropriately, our approach can deal with the motion of the object in a very wide area. Furthermore, multiple objects can be handled by managing multiple AHSs and processing multiple HKFs.

We show the validity of our approach using the synthetic data acquired simultaneously from the multiple virtual camera in a virtual environment (VE) and real data derived from a moving light display with walking motion. The results confirm that the model-based algorithm works well on the tracking of multiple rigid objects.

© 1997 Academic Press Limited

Soonki Jung\textsuperscript{1} and Kwangyun Wohn\textsuperscript{2}

Department of Computer Science,
Korea Advanced Institute of Science and Technology (KAIST),
373-1 Kusong-dong, Yusong-gu, Taejon 305-701, Korea
E-mail: \textsuperscript{1}skjung@dangun.kaist.ac.kr, \textsuperscript{2}wohn@cs.kaist.ac.kr

Introduction

Estimating the motion of a 3D object from a sequence of time-varying 2D images has been the subject of a rigorous investigation during the past decade. Most of the efforts have been largely restricted to rigid objects, even though many of the natural and man-made objects are non-rigid in the real world. Non-rigid motion can be generally classified...

Articulated motion estimation is usually divided into three subproblems: first, extracting and tracking features in 2D images; second, determining the 3D motion/depths of the primitives from these 2D correspondences; and thirdly, recognizing the 3D motion of articulated objects. First of all, it is important to choose a good feature so that the feature correspondence can be made as effortlessly as possible. Typically used features are points [3–6], line segments or edges [7–10], or deformable regions [11]. For the low-level feature extraction problem to be practically solvable, we use moving light displays (MLD) [12]. MLD-like input source has practical uses in various fields such as medical examination, motion analysis in sports, choreography of dance and ballet, and character animation [13,14]. See [15] for a survey of motion analysis from MLD.

Feature correspondence plays an important role in motion estimation. The problem has been extensively studied in radar imagery for target tracking as statistical data association [16–20]. Kurien discussed the key issues and concepts relevant to the design of practical multi-target tracking algorithms [19]. Blackman discussed some important issues and presented the methods applicable to the association and fusion of multiple sensor data [17]. Mathematically, the formulation of the data association problem leads to a generalization to the multi-dimensional matching problem, or equivalently, the multi-dimensional assignment problem [20]. And it is well known that the multi-dimensional matching problem is NP-complete, even under the assumption of zero false-alarm and unity detection probabilities [21]. Although the complexity of the data-association problem can sometimes be significantly reduced if the sampling rate of the sensors is increased, some ambiguity in motion correspondences is inherently unavoidable, especially in dynamic environments. The correspondence problem is still regarded as very difficult to solve. The theoretical base of the target tracking and surveillance is not directly applicable to tracking problems in articulated motion estimation. The two main reasons are: (i) occlusion—a moving feature may be partially or totally occluded by other body parts; and (ii) coherence of features—in radar imagery, a target represents an object, e.g. an aircraft, and it usually moved independently from the others [22]. In articulated motion, however, features from the same rigid body part undergo a coherent 3D motion.

There has been some previous work that deals with motion correspondence. Rashid established the correspondence by minimizing the difference between the expected position of each point and the actual position of the corresponding point [23]. The prediction of the point position is based on the observation that the velocity of the human or animal body varies smoothly, while the underlying structure is relatively fixed. Jenkin developed an algorithm that tracked the 3D MLD through time by fusing the two views of a scene [24]. These approaches, however, could not deal with the occlusion problem. Sethi and Jain suggested a method based on the smoothness of motion [25]. They firstly generated the trajectory for each point based purely on the nearest neighbor heuristics. Then, iteratively, they tried to gain the trajectories’ total smoothness by exchanging intermediate points until it was impossible to gain any further smoothness. Cheng and Aggarwal suggested a two-stage hybrid approach to the problem [3]; a sequential forward-searching algorithm (FSA), which extends all the surviving trajectories, and a batch-type, rule-based, backward-correcting algorithm (BCA). However, certain assumptions upon which their approaches are based are untrue in realistic situations.

Difficulties in feature correspondence cannot be overcome without a priori knowledge of the object shape. So model-based approaches to analyze articulated motion have been carried out in computer graphics and computer vision. O’Rourke and Badler used constraints on the human body model such as distance constraints, joint angle limits, and collision avoidance to refine the 3D joint positions [26]. Chen and Lee developed a 3D tracking algorithm with stick-figure models [4]. The algorithm firstly finds all possible 3D configurations of the body for each frame, and then finds the sequence of configurations that would best represent the walking motion. However, this algorithm is limited to the small possible configurations of the walking motion. Holt estimated the 3D motion of an articulated object from a monocular sequence of 2D perspective views, but he assumed that all the 2D joint trajectories were known in advance [5]. As a related work, Rohr introduced the model-based recognition of human movements with volumetric models, but the algorithm assumed that the observed person walked parallel to the image plane [9]. Leung and Yang proposed a posture recovery process by labeling the outline of a moving human body [10], but the algorithm assumed that there was only one moving human body in the scene, and produced a 2D human body stick figure.

The particular problem we would like to address in this paper is tracking MLDs of articulated objects using several
video cameras, and estimating the articulated motion and the pose of the articulated objects. To solve the tracking problem with the multi-view correspondence, we adopt a model-based approach. The articulated object is modeled using the volumetric hierarchical model, including the relative kinematic information such as the translational and the rotational parameters as well as the geometric data of each subject. These parameters are estimated and predicted using our newly developed technique named the hierarchical Kalman filter (HKF). An efficient data structure, the adaptive hierarchical structure (AHS), controls the order of the estimation sequence to facilitate the stability and accuracy of the HKF. Figure 1 depicts the block diagram of our approach.

The model-based approach used in this paper allows us to avoid not only the multi-view correspondence but also occlusion problem in the 3D feature tracking. We have implemented these strategies and experimented on the synthetic data acquired simultaneously from multiple virtual cameras in virtual environments, and real data derived from MLD with walking motion. In the next section we present the Kalman filter approach to the recovery of the 3D location and motion of a rigid segment. In later sections we propose the hierarchical Kalman filter and the adaptive hierarchical structure to extend the existing Kalman filter to the articulated object, demonstrate that our formulation works well through the several experiments, and give conclusions and details of future work.

3D Localization and Tracking of a Rigid Object

An articulated object consists of rigid parts (segments) connected together by joints with some kinematic constraints, such as a limited degree of freedom (DOF) in the rotational motion. Examples of objects which can be approximately modeled by articulated objects include human bodies, bicycles and robot arms. Firstly, we deal with the pose estimation of a rigid object from measurements through several cameras.

The 3D localization and tracking problem of the rigid object can be formulated as a state estimation problem in control system theory. The Kalman filter is well suited to this kind of problem, and has been adopted as a basic technique in this work. However, since the models turn out to be nonlinear, an extension of the Kalman filter known as the extended Kalman filter (EKF) is more appropriate for the problem. Especially in the case of significant nonlinearity in the measurement model, the estimates can be improved by the iterated EKF (IEKF). The derivation of generic IEKF can be found in [27]. An EKF approach to recover the 3D location and motion of a rigid object through a camera image was proposed by Wu et al. [28]. However, our formulation is more adequate, for several reasons. Firstly, our approach is capable of handling the ambiguities occurring in the motion correspondence. The ambiguity for each measurement is expressed as multiple measurements which are fused into the pose estimation during the iterated EKF loop.
Secondly, the multiple view approach alleviates the self-
occlusion problem and makes the 3D tracking of the rigid object possible. The measurements from multiple views are also fused into the measurement model of the EKF. Lastly, the quaternion representation to describe the orientation of coordinate system has several advantages over the conventional usage of direction cosines and Euler angles [29].

Camera model

An arbitrary object-coordinate frame can be described in terms of the translation vector \( t \) and the rotation matrix \( R \) with respect to the reference frame (3D world coordinates) [30]. In this paper, the rotation matrix \( R \) is represented by quaternion \( q \), which is the angular relation between co-ordinate systems (see Appendix A). One point \( P_0 \) in the object coordinate \( o \) with respect to the reference frame is

\[
\overrightarrow{P_0} = \overrightarrow{q} \otimes \overrightarrow{P_0} + t_0 = R_{01} \overrightarrow{P_0} + t_0
\]

where \( q^* \) denotes the conjugate of the quaternion \( q \).

Transformation from 3D world coordinate \( \overrightarrow{P_0} = (x_0, y_0, z_0)^T \) to image coordinate \( \overrightarrow{p} = (X_0, Y_0)^T \) through the camera \( c \) is defined by

\[
g_c : \overrightarrow{p} \rightarrow \overrightarrow{p}_c
\]

and \( g_c(\cdot) \) consists of the world-to-camera coordinate transformation function \( C(\cdot) \) and the camera-to-image coordinate transformation function \( T(\cdot) \) as follows:

\[
\overrightarrow{p} = C_{C}(\overrightarrow{p}) = q^* \otimes (\overrightarrow{p} - t_c) \otimes q_c
\]

\[
\overrightarrow{p}_c = T_c(\overrightarrow{p}) = \begin{bmatrix} X_c \\ Y_c \end{bmatrix}
\]

where \( q_c \) and \( t_c \) are the external parameters, and \( f_c \) is the internal parameter of the camera \( c \), respectively. When real imagery is involved, the camera parameters are measured using the camera calibration technique.

Dynamic model

Formulation of a target kinematic model requires the specification of the variables or the state of the target that needs to be estimated, and the equation, or dynamic model, that relates the target state from one posture in time to the next. The dynamic model is described by a difference equation:

\[
s_t = f(s_{t-1}) + n_{t-1}
\]

where \( f \) is a vector function describing the transition of the state vector from \( t - 1 \) to \( t \), and \( n \) is the random disturbance of the dynamic model.

We select thirteen variables constituting the position \( (p_x, p_y, p_z) \), the velocity \( (v_x, v_y, v_z) \), the angular position \( (q_w, q_x, q_y, q_z) \) and the angular velocity \( (w_x, w_y, w_z) \) for each rigid object in a reference frame.

\[
s = [s^T \ v^T \ q^T \ \Omega^T]^T
\]

and

\[
p = \begin{bmatrix} x \\ y \\ z \\ v_x \\ v_y \\ v_z \\ q_w \\ q_x \\ q_y \\ q_z \\ \Omega_x \\ \Omega_y \\ \Omega_z \end{bmatrix}
\]

where the angular velocity \( \Omega \) is a vector quaternion (see Appendix A), and we use the dummy element '0' for mathematical convenience. We represent the dynamic model for the target state by the following equation, under the assumption that the velocity and the angular velocity are constant in a small time interval, from Eqn (5):

\[
\dot{s}_t = f(s_{t-1}) = \begin{bmatrix} P_{p_{t-1}} \\ P_{v_{t-1}} \\ P_{q_{t-1}} \\ P_{\Omega_{t-1}} \end{bmatrix} = \begin{bmatrix} p_{t-1} + v_{t-1} \\ v_{t-1} \\ q_{t-1} \otimes q_{t-1} \\ \Omega_{t-1} \end{bmatrix}
\]

where the quaternion \( w_{t-1} \) is given by Eqn (A7).

The associated error covariance matrix is propagated to \( t \) by the following equation;

\[
P_{\dot{s}_{t-1}} = F_{t-1}P_{s_{t-1}}F_{t-1}^T + Q_{t-1}
\]

where \( F_{t-1} \) is the partial derivative matrix of \( f(s) \) with respect to \( s_{t-1} \) (see Appendix B), and \( Q_t \) is defined by
With the rigidity assumption of the segment, \( \tilde{p}_{m,0} \) is constant in time. So the 2D predicted measurement for the \( i \)th marker through the camera \( c \) is represented by the predicted state as follows:

\[
\tilde{z}_{m,c}(t) = h_{m,c}(\tilde{s}_{jt-1})
\]

(13)

Due to the error in the dynamic model, \( \tilde{z}_{m,c}(t) \) has the uncertainty given as, from the error covariance matrix of the predicted state:

\[
\hat{\Lambda}_{m,c}(t) = H_{m,c}(t)P_{jt-1}H_{m,c}^T(t)
\]

(14)

where \( H_{m,c}(t) \) is the partial derivative matrix of \( h_{m,c}(s) \) with respect to \( s_{jt-1} \) (see Appendix B). Due to sensor noise and the error in the dynamic model or the rigidity assumption, the prediction is not exactly matched to the actual measurement. At this stage, ambiguities may arise. In fact, resolving these ambiguities is the essence of motion correspondence. Given \( n \) observed measurements from the camera \( c \) at time \( t \) \( (z_c(t), j = 1, \ldots, n) \) with covariance matrices \( \{R_{c}(j)\}, j = 1, \ldots, n \), we use the Mahalanobis distance to decide which observed measurement matches the predicted measurement \( \tilde{z}_{m,c}(t) \) with the covariance matrix \( \hat{\Lambda}_{m,c}(t) \). The error covariance matrix of observed measurement, \( R(k) \), is defined by

\[
E[\eta \eta^T] = \begin{bmatrix} R(k) & 0 \\ 0 & 0 \end{bmatrix}
\]

As mentioned before, our tracking algorithm deals with the rigid object as the target instead of the markers. For a rigid target, a set of measurements is given as the 2D coordinates of the markers attached on the segment. As in Figure 2, we assume that the state of the segment is \( s_{jt-1} \) at time \( t-1 \) and the estimated state is \( \hat{s}_{jt-1} \) by the dynamic model, Eqn (8). When the \( i \)th marker on the segment is visible through the camera \( c \) at time \( t \), the image plane measurement of the marker is predicted as follows:

1. When we know the object coordinate \( \tilde{p}_{m,0} \) for the \( i \)th marker, the world coordinate \( \tilde{p}_{m,w} \) is given as follows, from Eqn (1):

\[
\tilde{p}_{m,w}(t) = W(s_{jt-1}, m_i) = q_{jt-1} \odot \tilde{p}_{m,0} \odot q_{jt-1}^* + p_{jt-1}
\]

(11)

2. For a camera \( c \), the world coordinate is transformed into the image coordinate according to the imaging model (2).

\[
\tilde{p}_{m,c}(t) = b_{c}(\tilde{p}_{m,w}(t))
\]

(12)

The Mahalanobis distance between the prediction of the \( i \)th marker and the \( j \)th observed measurement from the camera \( c \) is defined as

\[
d_{m,j,c}(t) = \tilde{r}_{m,j,c}(t) \Lambda_{m,j,c}(t)^{-1} r_{m,j,c}(t)
\]

(15)

where \( r_{m,j,c}(t) = z_c(t) - \tilde{z}_{m,c}(t) \) and \( \Lambda_{m,j,c}(t) = R_c(t) + \hat{\Lambda}_{m,c}(t) \). The variable \( d_{m,j,c}(t) \) is a scalar random variable following a Chi-squared distribution with \( n \) degrees of freedom, where \( n \) is the dimension of the measurement vector [37]. By looking up the Chi-squared distribution table, we can choose the appropriate threshold \( \epsilon \). If \( d_{m,j,c}(t) > \epsilon \), then the measurement \( z_c(t) \) is considered to be spurious and is rejected. Otherwise, the \( j \)th measurement is considered as a potential match of the marker. The ellipsoidal validation or the search volume is defined by the Mahalanobis criterion.

In an ideal situation, only a single measurement would fall inside the validation volume of each marker. However, when several measurements are near to each other, multiple
measurements may be validated to a marker. When the marker is invisible due to self-occlusion or out of the current field of view, the marker does not have its own measurement. In order to decide which measurement is to be assigned to the marker, the global optimization can be carried out by taking all markers and their plausible matches into account. The computation, however, will be very expensive. Several suboptimal algorithms to alleviate this difficulty have been studied [31,32].

We use a representative one from all the measurements in the validation as the measurement for each marker. In other words, the measurement is assumed to be the average of all the candidate measurements given by

$$z_{m,c}(t) \triangleq \frac{1}{N_{c}} \sum_{j=1}^{N_{c}} z_{j,c}(t)$$ (16)

where $N_{c}$ is the number of measurements in the validation volume, and as mentioned before, will be zero or more. When $N_{c}$ is zero, the measurement vector is null vector. Our formulation is similar to the combined weighted innovation in the idea of the JPDAF method proposed by Bar-Shalom and Fortmann [33]. The associated error covariance matrix is defined by

$$\Lambda_{m,c}(t) = \sum_{j=1}^{N_{c}} \Lambda_{m,j,c}$$ (17)

Consequently, the measurement vector for one rigid object a time $t$ is the augmented vector including all the measurements of the markers from all cameras.

$$z_{t} \triangleq \begin{bmatrix}
z_{m,1}(t) \\
\vdots \\
z_{m,S}(t) \\
z_{m,1}(t) \\
\vdots \\
z_{m,c}(t) \\
\vdots \\
z_{m,S}(t)
\end{bmatrix}, \text{ } 1 \leq c \leq S, \text{ } 1 \leq i \leq M$$ (18)

where $S$ is the number of cameras, and $M$ is the number of markers attached on the segment. The associated error covariance matrix, $\Lambda_{i}$, is defined by setting the diagonal elements corresponding to the measurement to its covariance matrix, $\Lambda_{m,c}(t)$, and the off-diagonal ones to zero, that is,

$$\text{Diag}[\Lambda_{i}] = \{\Lambda_{m,1,c}(t), \ldots, \Lambda_{m,j,c}(t), \ldots, \Lambda_{m,N_{c},c}(t)\}$$ (19)

**Iterated refinement**

Once the measurement vector $z_{t}$ has been found, the Kalman gain is computed:

$$K_{i} = P_{i|t-1} H_{i}^{T} \Lambda_{i}^{-1}$$ (20)

where $H_{i}$ is defined by

$$H_{i} \triangleq \begin{bmatrix}
H_{m,1}(t) \\
\vdots \\
H_{m,S}(t) \\
H_{m,1}(t) \\
\vdots \\
H_{m,c}(t) \\
\vdots \\
H_{m,S}(t)
\end{bmatrix}, \text{ } 1 \leq c \leq S, \text{ } 1 \leq i \leq M$$ (21)

The measurement residual is weighted by the Kalman gain matrix to generate a correction term and is added to the predicted state to obtain the updated state as:

$$s_{t} = \hat{s}_{i|t-1} + K_{i}(z_{t} - H_{i} \hat{s}_{i|t-1})$$ (22)

Finally, $P_{i|t-1}$ is used to update the covariance matrix $P_{i}$ by the following equation:

$$P_{i} = (1 - K_{i} H_{i}) P_{i|t-1}$$ (23)

The Kalman filter can be shown to be a recursive implementation of the weighted least-squares estimation, and the weighting factor is inversely proportional to the uncertainty of measurements. As shown in Eqn (17), the uncertainty of the measurement $z_{m,c}$ is proportional to the number of candidate measurements for the $i$th marker through the camera $c$. So the 3D posture estimation of the rigid object using the Kalman filter can be considered as fitting globally the marker positions of 3D model to the projected images from all cameras.

To alleviate the nonlinearity of the EKF, we apply the local IEKF to a single piece of sample data by refining the matching measurements, relinearizing the measurement equation, and then recomputing the better state estimates. This process may be iterated if desired.
Alternatively, articulated objects can be represented by the nested coordinate frame forming the hierarchical tree. The pose of the subpart can be described by the relative posture with respect to the coordinate frame of its parent. One example of the hierarchical model for articulated object is shown in Figure 4.

We will extend the solution for the rigid object to the articulated one. First, the states for all subparts other than the root are defined by the angular position and the angular velocity in the parent coordinate frame as follows:

$$s_i' = \begin{bmatrix} q_i^T \\ \Omega_i^T \end{bmatrix}, j \neq 0 \quad (24)$$

where the superscript $j = 0$ denotes the root and has the translational motion as well as the rotational motion. The dynamic model for each subpart $j$ but the root is similar to Eqn (8), i.e.

$$\dot{s}_i^{j-1} = f(s_i^{j-1}) = \begin{bmatrix} q_i^{j-1} \\ \Omega_i^{j-1} \end{bmatrix} = \begin{bmatrix} w_i^{j-1} @ q_i^{j-1} \\ \Omega_i^{j-1} \end{bmatrix} \quad (25)$$

However, in the measurement model, we cannot get the world coordinate $p_{m,p}^j$ directly from $W(s_i^{j-1}, m_i)$ in Eqn (11), but instead from the parent coordinates as follows:

$$p_{m,p}^j = P(s_i^{j-1}, m_i) \quad (26)$$

where $p(j)$ means the parent node of the subpart $j$ in the hierarchical tree. As mentioned by Hel-Or [34], the main problem with the parameter reduction method is the need to define the dependence of each measurement on all the free parameters during the estimation process. Since the motion of the subpart $j$ is described in the parent coordinate system as shown in Eqn (26), the measurement of the state $s_i^{j}$ is dependent on the motion of the ancestors of the subpart $j$ in the hierarchical structure. So the measurement $z_{m,c}^{j}(t)$ is given by
\[
\hat{z}_{m,c}^j(t) = h_{m,c}^j \left( \hat{z}_{d+1}^j, \hat{z}_{d+2}^j, \ldots \right) \tag{27}
\]

rather than

\[
\hat{z}_{m,c}^j(t) = h_{d+1}^j \left( \hat{z}_{d+1}^j \right)
\]

As shown in Eqn (27), the dependence of each measurement on all of its ancestors must be taken care of during the measurement estimation process. Furthermore, the higher the order of the nonlinearity of the dependence equation, the less stable the solution results.

It is possible to reduce the order of the nonlinearity by assuming that the state of the ancestors of the subpart \( j \) have already been determined during the estimation process of the subpart \( j \). With this assumption, Eqn (27) is rewritten as

\[
\hat{z}_{m,c}^j(t) = h_{m,c}^j \left( \hat{s}_{d+1}^j, \hat{s}_{d+2}^j, \hat{s}_{d+3}^j, \ldots \right) \tag{28}
\]

At this point, one point \( \hat{p}_{m,p(j)} \) with respect to the reference frame is given by

\[
\hat{p}_{m,w}(t) = \mathcal{W}_{p(j)} \left( \hat{p}(\hat{s}_{d+1}^j, m_{t}) \right) \tag{29}
\]

and \( \mathcal{W}_{p(j)} \) is defined as

\[
\mathcal{W}_{p(j)}(\hat{p}) = \hat{q}_{p(j)} \otimes \hat{p} \otimes \hat{q}_{p(j)}^* + \hat{p}_{p(j)}
\]

where the hats ' \( \hat{\cdot} \) ' denote the relation with respect to the reference frame, and \( \hat{q}_{p(j)} \) and \( \hat{p}_{p(j)} \) can be determined from the states of the ancestors of the subpart \( j \). The derivative matrix \( H_{m,c}^j(t) \) is also given by Eqn (B10). With the estimation process in the preorder traversal of the hierarchical tree, we can simply expand the estimation of the rigid segment to the articulated object. The main drawback of this approach, however, is that the error in the estimation process is propagated and amplified to the descendants.

We solve this problem by introducing the notion of the adaptive hierarchical structure (AHS). The root of the AHS is designated to the subpart whose state reveals the highest confidence, to minimize the global estimation error. We use the confidence measure of the subpart as the inverse of its error covariance matrix, which can be approximated by the trace of the covariance matrix as follows:

\[
E_{r_j} \triangleq \text{trace}(\hat{p}_{r(j)}^j)
\]

The hierarchical structure is maintained throughout the estimation process by rearranging the inter-part relationship. Figure 5 shows the binary tree representation, introducing the aggregator which accounts for the joint between two subparts. The aggregator includes the state of the subpart as well as the geometric information of the child nodes, thereby facilitating the root change process in the hierarchical structure. Table 1 depicts the algorithm to set an arbitrary node to the root in AHS.
Table 1. The algorithm to set an arbitrary node to the root in AHS

```c
void SetToRoot(Subpart *node)
{
    extern Subpart *Root;

    if (node == Root) return;
    if (node->parent != Root) SetToRoot(node->parent);
    node->prev->sibling = node->sibling;
    Root->parent->child = node;
    Root->sibling = node->child->sibling;
    node->child->sibling = Root;
    Root->parent = node;
    SwapState(node, Root);
    Root = node;
}
```

Figure 6. The DOF constraints between two subparts.

**Kinematic constraints of articulated object**

Articulated objects reveal their kinematic constraints between two connected subparts. The constraints can be simply modeled in the measurement model with zero uncertainty. This idea has been used in [27,34]. In our approach, we use the DOF constraints for the kinematic constraints. In the previous section we modeled all joints with three DOF for simplicity and greater generality of the formulation. The DOF constraints can be viewed as measurements with very small uncertainties. When a joint \( j \) has only one DOF and the directional cosine vector is \( \vec{n} \), as shown in Figure 6, we can obtain two orthonormal vectors for \( \vec{n} \) given by

\[
\vec{n} \cdot \vec{v}_i = 0, \text{ for } i = 1, 2
\]

Similarly, when a joint has two DOF and the two directional cosine vectors are \( \vec{n}_1 \) and \( \vec{n}_2 \), the quaternion will be represented by a spherical linear combination of \( \vec{n}_1 \) and \( \vec{n}_2 \), and we can get the orthonormal vector \( \vec{v}_i \) by

\[
\vec{n}_i \cdot \vec{v} = 0, \text{ for } i = 1
\]

During the estimation process, the constraints are given as

\[
\hat{E}(q) \cdot \vec{v}_i = 0
\]

and

\[
\Omega \cdot \vec{v}_i = 0
\]

where \( \hat{E}(q) \) represents the rotation axis of the quaternion \( q \). \( E(q) \) is given as

\[
\hat{E}(q) = \frac{\theta}{\sin(\theta / 2)} \vec{q}, \text{ where } \theta = 2\arccos(q_w)
\]

and \( \vec{q} \) denotes the vector quaternion of \( q \). We have added the constraints (33) and (34) in the measurement model of (28). The derivatives of (33) and (34) with respect to \( s \) are given by

\[
\frac{\partial \left( \hat{E}(q) \cdot \vec{v} \right)}{\partial s} = \begin{bmatrix} 0^T & 0^T & \frac{\theta q_w - 2\sin(\theta / 2)}{\sin^2(\theta / 2)} (1 - 4q_w^2) \end{bmatrix} \frac{\theta}{\sin(\theta / 2)} \vec{v}_i \begin{bmatrix} 0^T \end{bmatrix}
\]

and

\[
\frac{\partial \Omega \cdot \vec{v}}{\partial s} = \begin{bmatrix} 0^T & 0^T & 0^T \end{bmatrix} \begin{bmatrix} 0^T \end{bmatrix}
\]

where \( 0_n \) means zero vector with \( n \) rows.

Since we use the quaternion to represent the rotation, we have added the constant \( \|q\| = 1 \), i.e., \( q^T q - 1 = 0 \) in (13) and (28) as the additional measurement. The derivative of \( q^T q - 1 \) with respect to \( s \) is given by

\[
\frac{\partial}{\partial s} \left( q^T q - 1 \right) = \begin{bmatrix} 0^T & 0^T & 2q^T & 0^T \end{bmatrix} \begin{bmatrix} 0^T \end{bmatrix} \begin{bmatrix} 2q^T \end{bmatrix}
\]

where \( j = 0 \) and \( j \neq 0 \).

**Visibility test**

In the 3D tracking process, the markers may be occluded by other subparts or may disappear by going out of the field of view. Without considering the occlusion problem, the measurements are easily mismatched with the invisible markers. The prediction of the visibility for the markers through all cameras is important to improve the performance of the matching process. Firstly, we get the predicted state with the dynamic model of Eqn (8). The visibility test is performed with all the markers for each view. From the test, we can classify the measurements for a marker into several cases:
(i) occlusion – the marker may be occluded by other subparts, or self-occluded by rotation of the subpart; (ii) disappearance – the marker is invisible by moving out of the field of view in the next frame; (iii) appearance – the previously unseen marker comes into view; and (iv) absence – when the marker which could be present is not measured due to the sensory error or the prediction error. When many cameras are used to cover a very wide area, we can decide which cameras are detectable for each marker by a test of the field of view.

In order to determine the visibility of all the markers efficiently, we generate an image with the human body model
using the liem buffer [35,36], which includes the marker's ID for each pixel during the z-buffer rendering process.

Experimental Results

In this section we present the results of running the algorithm on two experiments, one synthetic image sequence and one real MLD data.

Example 1: synthetic motion

Firstly, the simulated experimental setting consists of the hierarchical human model with 11 rigid subparts and 24 markers, and the five virtual cameras in virtual environments as shown in Figures 7 and 8.

In order to study the validation of the HKF, we generated a simulation of a human walking using a volumetric model...
created by forward kinematics with constant translational and angular velocity per each frame, and have extracted the MLD sequence shown in Figure 9.

With the standard HKF approach, the error of the estimated posture is propagated and amplified to the descendants during the estimation process of each subpart. Figure 10 shows the experimental results of the motion estimation with the fixed root node (pelvis) in the hierarchical structure. At the 11th frame, several markers of the root node are occluded with two arms, and the estimated posture is very unstable. This error resulted in a mismatch of the other subparts in the following frames.

Figure 11 shows the final results of the estimation process using HKF and AHS. In this case, the root node of AHS has the transit (pelvis \(\rightarrow\) chest \(\rightarrow\) pelvis) for a total of 35 frames. As shown in Figure 12, the position trajectory of the most unstable subpart (e.g. the leaf node in the AHS) is very similar to the original synthetic one. Figure 13 depicts the relative errors (the norm of the error vector divided by the norm of the true vector) of the translational and rotational posture for several subparts, and Table 2 shows the average errors are about 3% for the position and about 0.072% for the orientation, respectively. We can also see from the table that the error in the estimation process is not amplified to the descendants of the hierarchical structure.

**Example 2: real human motion**

We present here the results of the experiment on real human motion. This experiment was conducted with a

<table>
<thead>
<tr>
<th>Subpart</th>
<th>Position</th>
<th>Orientation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pelvis</td>
<td>0.045830</td>
<td>0.000899</td>
</tr>
<tr>
<td>Head</td>
<td>0.012229</td>
<td>0.000246</td>
</tr>
<tr>
<td>R.u.arm</td>
<td>0.021657</td>
<td>0.000663</td>
</tr>
<tr>
<td>R.thigh</td>
<td>0.050479</td>
<td>0.000811</td>
</tr>
<tr>
<td>R.forearm</td>
<td>0.024269</td>
<td>0.001101</td>
</tr>
<tr>
<td>R.calf</td>
<td>0.030533</td>
<td>0.000532</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Subpart</th>
<th>Position</th>
<th>Orientation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chest</td>
<td>0.032304</td>
<td>0.000547</td>
</tr>
<tr>
<td>L.u.arm</td>
<td>0.017904</td>
<td>0.000699</td>
</tr>
<tr>
<td>L.thigh</td>
<td>0.049578</td>
<td>0.000767</td>
</tr>
<tr>
<td>L.forearm</td>
<td>0.024327</td>
<td>0.001272</td>
</tr>
<tr>
<td>L.calf</td>
<td>0.028051</td>
<td>0.000397</td>
</tr>
</tbody>
</table>
preliminary optical motion capture system. The multiview system we used consists of four synchronized cameras. The image resolution is $640 \times 480$ pixels, and the subject of our capture system has 24 markers attached to the body, as shown in Figure 14. The camera system was calibrated using Tsai's calibration technique [37] using a non-coplanar set of points. Figure 15 depicts the calibration template and the calibration result for the first view.

The volumetric human body model consists of 11 subparts forming the hierarchical structure (like Figure 4) with 24 attached markers, as shown in Figure 16. The geometric size of each subpart and the positions of markers should be a predetermined match with the real subject's. Figure 17 and Figure 18 demonstrate the sample image sequence and the superimposed image sequence of the synthetic actor animated by the estimated motion. The 3D tracking and motion estimation algorithm correctly worked on real imagery.

In the current work, the image sequences were recorded by using four video recorders and then captured off-line while playing back frame-by-frame. We used a Panasonic
AG-DS850 and a Pentium PC for this process. The whole system was implemented on a Silicon Graphics Onyx Reality Station. The computational complexity of each step is as follows: (i) feature extraction – the MLD data is extracted with simple local-thresholding and image labelling algorithm. This routine can be executed in real-time by using specialized image processing hardware. (ii) Prediction – the posture of the human body at the next time is predicted by several matrix multiplication, as shown in the dynamic model 5. (iii) Visibility test – the visibility of each marker from a camera is tested by one image rendering and one frame buffer scanning process. (iv) Searching 2D measurements and updating states – the Kalman filter loop for each segment is performed. (v) Animation – the estimate results control an animated character. The time complexity of each step in Example 2 is shown in Table 3. The time for estimation1 (for pelvis and chest) is longer than estimation2 (for other subparts) because of the number of markers. It takes 374.237 ms to estimate the whole body by a sequential processing (the current implementation).

Since our tracking algorithm has been formulated as an iterative estimation procedure, and several computation steps can be executed independently, it is feasible to carry out the entire computation in real time. Figure 19 shows a possible scheduling scheme on four multiprocessors. All of the computation steps have a partially ordered dependency.
Table 3. The average time complexity of each step in Example 2.

<table>
<thead>
<tr>
<th>Computing step</th>
<th>Unit time (ms)</th>
<th>Number of views(v), iteration(i), subparts(s)</th>
<th>Total time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature (F)</td>
<td>36.427</td>
<td>4 (v)</td>
<td>145.708</td>
</tr>
<tr>
<td>Prediction (P)</td>
<td>0.509</td>
<td>–</td>
<td>0.509</td>
</tr>
<tr>
<td>Visibility (V)</td>
<td>42.479</td>
<td>4 (v)</td>
<td>169.916</td>
</tr>
<tr>
<td>Estimation1 (E1)</td>
<td>4.762</td>
<td>2–4(i), 2(s)</td>
<td>28.572</td>
</tr>
<tr>
<td>Estimation2 (E2)</td>
<td>1.573</td>
<td>1–3(i), 9(s)</td>
<td>28.314</td>
</tr>
<tr>
<td>Animation (A)</td>
<td>1.218</td>
<td>–</td>
<td>1.218</td>
</tr>
<tr>
<td>Whole step</td>
<td>374.237</td>
<td>–</td>
<td>374.237</td>
</tr>
</tbody>
</table>

Figure 19. An example of job scheduling on four multiprocessors (by software simulation).

like $F \rightarrow V \rightarrow E$. The process group $E$, consists of a sequence of the estimation processes in the breadth-first sorting of the AHS. Furthermore, since these three process groups are also independent, the computation time of the whole system can be reduced to about 40 ms with pipeline processors.

Conclusion

In this work we have developed a system for the estimation of the 3D motion and posture of an articulated volumetric object from a sequence of multiple perspective views. First, we discussed the Kalman filter for 3D localization and tracking of a rigid object. To avoid the multiview correspondence problem and the occlusion problem in the 3D feature tracking, we took the model-based approach. Our approach is able to handle the ambiguities that often occur in motion correspondence. Furthermore, the multiple view approach alleviates the self-occlusion problem and makes the 3D tracking of a rigid object possible.

We extended the solution for one rigid object to the articulated object with the newly proposed hierarchical Kalman filter. Our approach is based on parameter reduction methods to make the pose of the object equal to the degree of freedom of the object. The AHS controls the order of the estimation sequence to facilitate the stability and accuracy of the HKF. With the volumetric models, we used related kinematic information such as DOF constraints and visibility information, which are important to improve the performance of the matching process.

Two experiments performed on the synthesized human walking motion, and the other, on an actual human, showed the validity of our approach. We believe that the system is useful for real-time optical motion capture and for the full-body interface in a virtual reality system.

References


Appendix A: Rotational Motion with Constant Angular Velocity

A unit quaternion \( q = \begin{bmatrix} q_w \\ q_x \\ q_y \\ q_z \end{bmatrix} = \begin{bmatrix} \cos \frac{\theta}{2} \\ \frac{\sin \frac{\theta}{2}}{2} u_x \\ \frac{\sin \frac{\theta}{2}}{2} u_y \\ \frac{\sin \frac{\theta}{2}}{2} u_z \end{bmatrix} \) where \( u = (u_x, u_y, u_z) \) is the direction cosines of an axis of rotation, and \( \theta \) is the angle about that axis that describes the rotation of the object coordinate system from its initial orientation to its final orientation. The finite rotation in space is represented by
\[ \alpha' = q \otimes \alpha \otimes q^* \]  

where \( \alpha' \) and \( \alpha \) are vector quaternions of which the first elements are zero, and \( \otimes \) means the quaternion multiplication [9].

Differentiating Eqn (A2), the quaternion propagates in time according to the differential equation

\[ \dot{q} = \frac{1}{2} \Omega \otimes q = \frac{1}{2} \hat{\Omega} q \]  

where \( \Omega \) is vector quaternion and \( \hat{\Omega} \) is matrix representation for the quaternion multiplication [9]. When \( \Omega \) is constant for unit time, the solution to Eqn (A3) is

\[ q_{t+1} = e^{\hat{\Omega}/2} q_t \]  

By Rodrigues' formula [27], the power series expansion for the matrix exponential can be reduced to

\[ q_{t+1} = \left[ \cos \left( \frac{\Omega_t}{2} \right) I_4 + \frac{\sin \left( \frac{\Omega_t}{2} \right)}{\Omega_t} \hat{\Omega} \right] q_t \]  

and the above equation is represented by

\[ q_{t+1} = w_t \otimes q_t \]  

where the quaternion \( w_t \) is the quaternion representation of the vector quaternion \( \Omega_t = [\omega_x, \omega_y, \omega_z]^T \) defined by

\[ w_t = \begin{bmatrix} \cos(d/2) \\ \frac{d}{2} \sin(d/2) \omega_x \\ \frac{d}{2} \sin(d/2) \omega_y \\ \frac{d}{2} \sin(d/2) \omega_z \end{bmatrix}, \quad d = |\Omega_t| \]

Appendix B: Linearization of the Equations

The state transition and measurement observation functions are nonlinear. For the dynamic model (8), the partial derivative matrix of \( f(s) \) with respect to \( s_{t-1} \) is given by

\[ F_{t-1} = \begin{bmatrix} I_3 & I_3 & 0 & 0 \\ 0 & I_3 & 0 & 0 \\ 0 & 0 & \frac{\partial q}{\partial q'} & \frac{\partial q}{\partial \Omega} \\ 0 & 0 & 0 & I_3 \end{bmatrix} \]  

where the partial derivatives for the rotational posture and the rotational velocity with respect to \( q_{t-1} \) and \( \Omega_{t-1} \) are as follows:

\[ \frac{\partial q}{\partial q_{t-1}} = \cos(\Omega_{t-1}/2) I_4 + \frac{\sin(\Omega_{t-1}/2)}{|\Omega_{t-1}|} \Omega_{t-1} \]  

\[ \frac{\partial q}{\partial \Omega_{t-1}} = -\frac{\sin(\Omega_{t-1}/2)}{|\Omega_{t-1}|} q_{t-1}, \Omega_{t-1} \]  

\[ + \frac{\left( \cos(\Omega_{t-1}/2) - \frac{\sin(\Omega_{t-1}/2)}{|\Omega_{t-1}|} \right)}{2|\Omega_{t-1}|^2} \Omega_{t-1} q_{t-1}, \Omega_{t-1} \]

and \( \hat{\alpha} \) and \( \hat{\alpha} \) are matrix representation for the quaternion multiplication [29].

The partial derivative matrix of the observation function for only one measurement (the \( i \)th marker through the camera \( c \)) is represented by Eqn (13):

\[ H_{m,c}(t) = \frac{\partial h_{m,c}(s)}{\partial s} \]

First, the partial derivative matrix of \( W(\cdot) \) is given by, from Eqn (11):

\[ \frac{\partial W(s, m_i)}{\partial s} \]  

and

\[ \frac{\partial \tilde{p}_{m,w}(t)}{\partial q} \]

where \( \tilde{p}_{m,w} = [p_x, p_y, p_z]^T \) and \( q_{el-1} = [q_w, q_x, q_y, q_z]^T \). The partial derivative of \( C(\cdot) \) is given by Eqn (3):
\[ \frac{\partial C_c(p_w)}{\partial p_w} \bigg|_{p_w(t)} = R_c^T \]  

(B7)

The partial derivative matrix of \( T_c(\cdot) \) is given by Eqn (4):

\[ \frac{\partial T_c(p_c)}{\partial p_c} \bigg|_{p_w(t)} = \begin{bmatrix} f_c & 0 & -f_c x_{ic} \\ z_{ic} & f_c & -f_c y_{ic} \\ 0 & z_{ic} & -f_c y_{ic} \end{bmatrix} \]  

(B8)

As a result, the partial derivative matrix of the observed function, Eqn (B4), is given by Eqns (B5), (B7) and (B8):

\[ H_{m,c}(t) = \frac{\partial T_c \circ C_c \circ W}{\partial \mathbf{s}}(\hat{s}_{j-1},m_i) \]  

(B9)

\[ = \frac{\partial T_c(p_m(t))}{\partial p_c} \frac{\partial C_c(p_{m,w}(t))}{\partial p_w} \frac{\partial W}{\partial s}(\hat{s}_{j-1},m_i) \]

In the case of an articulate object, the measurement model for subpart \( j \) is given by Eqns (3), (4), (26) and (29).

\[ h_{m,c}^j(\hat{s}_{j-1}) = T_j(C_j(W_{p(j)}(P(\hat{s}_{j-1},m_i)))) \]  

where \( j \neq 0 \).  

(B10)

The partial derivative matrix of the observed function for subpart \( j \) is given by

\[ H_{m,c}(t) = \frac{\partial T_c \circ C_c \circ W_{p(j)}}{\partial s}(\hat{s}_{j-1}^j,m_i) \]

\[ = \frac{\partial T_c(p_m(t))}{\partial p_c} \frac{\partial C_c(p_{m,w}(t))}{\partial p_w} \frac{\partial W_{p(j)}}{\partial s}(\hat{s}_{j-1}^j,m_i) \]

(B11)

where \( \frac{\partial T_c(p_m(t))}{\partial p_c} \) and \( \frac{\partial C_c(p_{m,w}(t))}{\partial p_w} \) are defined by

\[ \frac{\partial T_c(p_m(t))}{\partial p_c} \]  

and \( \frac{\partial C_c(p_{m,w}(t))}{\partial p_w} \)

Eqns (B8) and (B7), respectively. The derivative matrix of \( W_{p(j)}(\cdot) \) is given by

\[ \frac{\partial W_{p(j)}}{\partial p_{\alpha(j)}}(\hat{\mathbf{p}}_{m_{\alpha(j)}(t)}) = R_{p_{\alpha(j)}} \]  

(B12)

where \( R_{p_{\alpha(j)}} \) denotes the matrix representation for the rotation \( \hat{q}_{p(j)} \) with respect to the reference frame. The derivative matrix of \( P(\mathbf{s},m_i) \) is given by

\[ \frac{\partial P(\mathbf{s},m_i)}{\partial \mathbf{s}} \bigg|_{\hat{s}_{j-1}} = \left[ \begin{array}{c} \frac{\partial \hat{\mathbf{p}}_{m_{\alpha(j)}(t)}}{\partial \mathbf{q}} \\ \mathbf{0}_3 \end{array} \right] \]  

(B13)

where \( \frac{\partial \hat{\mathbf{p}}_{m_{\alpha(j)}(t)}}{\partial \mathbf{q}} \) is defined by Eqn (B6) with \( \hat{\mathbf{p}}_{m_{\alpha(j)}} = [p_x, p_y, p_z]^T \) and \( \hat{\mathbf{q}}_{j-1} = [q_x, q_y, q_z]^T \).