Video-Guided Motion Synthesis Using Example Motions

MIN JE PARK
Electronics and Telecommunications Research Institute
MIN GYU CHOI
Kwangwoon University
YOSHIHISA SHINAGAWA
University of Illinois at Urbana-Champaign
and
SUNG YONG SHIN
Korea Advanced Institute of Science and Technology

Video taken from a single monocular camera is the most common means of recording human motion. In this article, we present a practical, semiautomatic method for synthesizing a human motion that is guided by such video. After preprocessing an input video, we select a precaptured motion clip called a *reference motion* from a motion library. We then compute the sequence of body configurations of a virtual character by deforming this motion, according to spacetime constraints derived from a sequence of 2D features in the input video. Experimental results show that our method can synthesize highly dynamic motions, such as kicking and header motions of soccer players. We also showed the potential of our scheme as a new paradigm for motion capture, that is, capturing motions from videos taken with a monocular camera, even outside a motion capture studio.

Categories and Subject Descriptors: I.3.7 [**Computer Graphics**]: Three-dimensional Graphics and Realism—*Animation*; G.1.6 [**Numerical Analysis**]: Optimization—*Nonlinear programming*

General Terms: Algorithms, Experimentation

Additional Key Words and Phrases: Computer animation, human motion reconstruction, motion reuse, nonlinear optimization animation

1. INTRODUCTION

Since monocular videos are the most common medium for archiving human motions, many approaches have been proposed to capture human motions from videos for various purposes [Barron and Kakadiaris

This work was supported by the Korea Science and Engineering Foundation (KOSEF) and the Brain Korea (BK) 21 project of the Ministry of Education and Human Resources Development (MOE). M. J. Park was also supported by the Ministry of Information and Communication (MIC). M. G. Choi was supported by the Research Grant of Kwangwoon University in 2005.
Authors' addresses: M. J. Park, Electronics and Telecommunications Research Institute, 161 Gajeong-Dong, Yuseong-Gu, Daejeon-Gwangyeokshi, South Korea 305-350; M. G. Choi, Kwangwoon University, 447-1, Wolgye-Dong, Nowon-Gu, Seoul Korea 139-701; Y. Shinagawa, University of Illinois at Urbana-Champaign, 412 South Peoria St., Chicago, IL 60607; S. Y. Shin, Korea Advanced Institute of Science and Technology, 373-1 Guseong-Dong, Yuseong-Gu, Daejeon 305-701, Republic of Korea; email: syshin@jupiter.kaist.ac.kr.

2000; Bregler and Malik 1998; Kakadiaris and Metaxas 1996; Noma et al. 1999; Rehg and Kanade 1994; Sidenbladh et al. 2000a; Sminchisescu and Triggs 2001; Taylor 2000; Wren et al. 1997; Zheng and Suezaki 1998]. The computer vision community has been actively studying issues in motion capture and has made great advances. However, it is hard, if not impossible, to reconstruct a unique 3D motion from a monocular video due to the loss of depth information. In this article, we present a new method for synthesizing a human motion that is guided by a monocular video.

Previous methods for reconstructing a motion from monocular videos can be classified into two categories: manual and automatic methods. An example of the former can be observed in live sports TV broadcasting. Here, the 3D postures of players at important moments are rebuilt interactively from their corresponding 2D images to show snapshots from various views, using authoring tools such as Filmbox™ and Poser™ [Filmbox 2006; Poser 2006]. Features such as joint positions are manually specified at each video frame, and a posture is chosen from the possible candidates by manual interaction assuming that the length of each segment of the human body is known. There are several problems in this approach. First, it may require a great deal of user interaction to make a plausible posture. Second, the temporal coherence between consecutive frames cannot be guaranteed without postprocessing, such as smoothing and interpolation. This problem becomes conspicuous when there are occlusions, that is, when some joints are hidden at some video frames. Based on computer vision techniques, automatic methods track features to identify postures in the video sequence. However, such methods are still error-prone, that is, there are no methods that yield correct answers to any inputs, particularly when the inputs are TV broadcast sports videos that contain various motions with cluttered backgrounds. For example, the method in Pavlovic et al. [1999] requires that each frame of a gait sequence be sufficiently close. The approach in Brand [1999] requires silhouette extraction, which is known to be a difficult task when the background is cluttered. Regardless of the category of method we employ, either manual or automatic, users are faced with other problems: First, the length of each segment of a human body is not known in many cases, such as in sports videos. Second, errors in joint positions are inevitable, which amplifies the error in posture estimation.

To overcome these difficulties and produce realistic human motions from monocular videos, we solicit additional help from a precaptured motion. In other words, we exploit a densely sampled reference motion to produce a highly dynamic motion from a video of a low sampling rate. From this point of view, our approach is a motion synthesis scheme, rather than a motion reconstruction scheme. The availability of a high-quality motion similar to the target motion is the major premise of our approach. We solve the inherent difficulties of 3D reconstruction by transforming a reference motion from the library so that the postures projected onto the image plane match those in the video sequence. Segment lengths are also automatically optimized during the transformation. Given the exact positions of the joints in the video, our method can yield a motion that very closely approximates the motion in the video. Using the reference motion, we aim at generating plausible motions while allowing considerable errors in input joint positions. Use of the reference motion significantly reduces the amount of user interaction, and enables synthesis of smooth motions. We apply the proposed method to real sports videos to validate its effectiveness.

Our motion synthesis scheme assumes that a rich repertoire of human motions is available. In theory, a motion library containing all possible motion models is not always available, since human motions are too diverse to be accommodated in the library. We take a practical approach, assuming that we know what to synthesize in advance. For example, suppose that we are to synthesize the shooting motions of a soccer game recorded in a video. After acquiring a library of appropriate live-captured motion clips for shooting, we synthesize the shooting motions of players in the video.

Typically, motion capture is a time-consuming task performed in a motion capture studio, that is, a carefully controlled environment equipped with well-calibrated motion capture devices and facilities.

Skilled motion performers are also sometimes needed to capture high-quality motions. Moreover, post-processing of raw captured data is indispensable for clean motions. Under a mild assumption, our motion synthesis scheme can be a simple, effective alternative to the traditional method of motion capture. In other words, our scheme can be employed to derive a desired motion from the corresponding reference motion selected from the library, in accordance with the 2D features of an image stream captured with a monocular video camera, even outside a motion capture studio.

2. RELATED WORKS

2.1 Motion Reconstruction

Human motion reconstruction has been extensively studied in computer vision. Gavrila [1999], Moeslund and Granum [2001], and Wang et al. [2003] gave excellent surveys on the approaches to reconstructing human motions from image sequences. In this article, we classify these into two major categories, according to the availability of 3D body models: *model-based* and *feature-based* approaches.

Most approaches in the former category require stereoscopic images or multiple views [Delamarre and Faugeras 2001; Kakadiaris and Metaxas 1996; Leung and Yang 1995; Plaencker and Fua 2001; Rehg and Kanade 1994; Song et al. 2003]. In these approaches, 3D body models are fitted to images. A body is usually modeled by a tree of cylinders or rectangles. For example, Delamarre and Faugeras [2001] used virtual forces to fit a 3D articulated model to a 3D human body that was reconstructed by stereo matching. Starck and Hilton [2003] also computed body postures, in this case by using a human body model that consisted of 8,000 polygons and 17 articulated joints. Sminchisescu and Triggs [2001] pointed out the difficulties in recovering a 3D human body configuration from a single video stream due to ambiguity and occlusion problems. To alleviate the problem due to local minima, they incorporated covariance-scaled sampling into numerical optimization. Drummond and Cipolla [2001] represented the human body with several 3D quadrics. They estimated the rigid motion of each quadric separately with a statistical model, and then propagated the statistics of each quadric through a kinematic chain to obtain maximum *a posteriori* estimates of the pose of the entire structure. Recently, Kirk et al. [2005] proposed a scheme for accurately estimating skeleton topology, as well as segment lengths, from marker datasets acquired by an optical motion capture system. From this information, they reconstructed the orientation for each segment over time. However, these methods require multiple views of the scene, except the approach in Sminchisescu and Triggs [2001], which is designed specifically for hand movements. In our method, we aim at synthesizing motions from monocular video sequences such as TV broadcast sports video sequences.

Feature-based approaches can be classified into two groups, according to the availability of 3D motions. In Bregler and Malik [1998], Bregler et al. [2004], Demirdjian et al. [2003], Pavlovic et al. [1999], and Wren et al. [1997], 3D reconstruction of human motions relies purely on video sequences. For example, Demirdjian et al. [2003] took advantage of the fact that a multibody articulated motion space can be approximated by a linear manifold estimated from previous body poses which are sufficiently similar to each other. Pavlovic et al. [1999] proposed a method for tracking gait motions in the framework of a switching linear dynamic system (SLDS). The parameters of SLDS are learned from video data. Hence, a learning procedure is needed for various viewing directions. Bregler and Malik [1998, 2004] also estimated the motion from a video sequence taken from one or more cameras. They used a brightness constancy condition, $I(x, y, t) = I(x + dx, y + dy, t + dt)$, on the optical flow, while representing the projection of an articulated model by *twist* and *exponential maps*, and then reconstructed its 3D configuration. Wren et al. [1997] modeled a human body with several blobs and tracked the 3D position of each blob with a statistical dynamic model. They processed single/stereo video streams in real-time.

In Brand [1999], Howe et al. [2000], Morris and Rehg [1998], and Sidenbladh et al. [2000b], 3D motion tracking is formulated as an inference problem, relying on prior knowledge of 3D human motions. Howe et al. [2000] expanded the 2D tracking method in Morris and Rehg [1998] to track an entire body consisting of 20 body parts, and built each body part model as the weighted average of previous frames. They solved the underdetermined problem of 3D reconstruction from 2D tracking data by referring to training data gathered in a professional studio, that is, the probability of a short motion is determined according to a mixture-of-Gaussian probability model built from training data. Sidelbladh et al. [2000b] also used a learned pattern of walking motion in their Bayesian framework. Brand [1999] mapped input 2D silhouettes onto 3D body poses in motion capture data. Tian et al. [2005] trained a Gaussian process latent variable model with synthetic data to estimate upper body poses from 2D silhouettes. Grochow et al. [2004] presented a style-based inverse kinematics scheme based on a learned model of human poses. They used this idea to allow a user to select the most plausible 3D poses semiautomatically from known 2D features.

In Bregler and Malik [1998], Bregler et al. [2004], and Demirdjian et al. [2003], the postures in consecutive frames are required to be sufficiently similar. Due to restriction on the number of HMM states, the method of Brand [1999] sometimes infers the postures that may be different from those in the video. The methods of Grochow et al. [2004], and Howe et al. [2000] are mainly for reconstructing postures, rather than motions, since neither explicitly addresses temporal coherency nor root trajectory. Other methods are rather dedicated to specific movements such as walking [Pavlovic et al. 1999; Sidenbladh et al. 2000b], or need additional images from another viewpoint to resolve depth ambiguity [Wren et al. 1997].

There is yet another line of research which bypasses the automatic extraction of features and concentrates on efficient extraction of 3D information from 2D features that are specified manually. Taylor [2000] recovered the 3D configuration of an articulated structure whose ratios of segment lengths are known from manually specified joint positions, while considering the foreshortening of segments in the image. Zheng and Suezaki [1998] introduced a model-based approach to acquire motions of an articulated model from a single video stream. They selected several keyframes to recover the configurations of the model and interpolated them to obtain a motion. Urtasun et al. [2005] employed principal component analysis to build a motion model from motion capture data, and then recovered golf swing from a monocular video. Difranco et al. [1999] proposed an offline algorithm to estimate the maximum *a posteriori* trajectory from 2D measurements that were subject to a number of constraints, such as a kinematic model and joint angle limits. Barron and Kakadiaris [2000] extended this idea to estimate anthropometrical data from a single image. Liebowtiz and Carlsson [2001] presented an algorithm for the 3D reconstruction of a dynamic articulated strucure from uncalibrated multiple views. They exploited constraints associated with the strucure, in particular, the conservation of segment lengths over time. Gleicher and Ferrier [2002] compared previous approaches for reconstructing human motions from visual observations and described the inherent difficulties in motion reconstruction. We try to minimize user interactions while overcoming the difficulties in obtaining convincing motions automatically from monocular videos.

2.2 Motion Reuse

Due to the current success of motion capture technology, there has been a vast amount of literature reported in motion capture and reuse. We concentrate on only those works that are directly related to our scheme.

Witkin and Kass [1988] proposed a spacetime constraint approach to produce the optimal motion that satisfies a set of user-specified features. Cohen [1992] developed a spacetime control system that allows a user to interactively guide a numerical optimization so as to find an acceptable solution in a

feasible amount of time. Bruderlin and Williams [1995] introduced the concept of displacement mapping to alter a motion while preserving its details, and Witkin and Popović [1995] presented a motion warping technique for the same purpose. Rose et al. [1996] adopted this approach to generate a smooth transition between motion clips. Gleicher [1997] simplified the spacetime problem by removing the physics-related aspects from the objective function and constraints. He also applied this technique for motion retargetting [Gleicher 1998]. For interactive performance, Lee and Shin [1999] combined a hierarchical curve fitting technique with a new inverse kinematics solver for adaptively refining a motion to meet spacetime constraints. Safonova et al. [2004] showed that the spacetime problem can be solved efficiently by projecting motions onto a low-dimensional, behavior-specific space.

Popović and Witkin [1999] introduced a novel algorithm that takes dynamics into consideration. They simplified a complex dynamic system, without losing the fundamental dynamic properties of motion. Tak et al. [2000] proposed a motion balance filter that postprocesses the edited motion to keep the dynamic balance by using the notion of a zero-moment point (ZMP). Liu and Popović [2002] presented a method for the rapid prototyping of a realistic character motion using a set of dynamic constraints. They exploited a pattern of angular momentum transfer acquired from biomechanics data for realistic motion generation. Fang and Pollard [2003] introduced an efficient method to compute the first derivatives of objective functions and constraints in order to accelerate the optimization process for physically-based motion generation. Yamane et al. [2004] proposed a method for realistic character animation using data-driven, constraint-based inverse kinematics.

Video-guided motion synthesis by rearrangement is yet another line of related work. Approaches in this category generate motions similar to an input video by motion rearrangement, that is, the cut-and-paste of either short motion clips or fragments of large motion clips [Lee et al. 2002; Ramanan and Forsyth 2003; Sidenbladh et al. 2002]. Sidenbladh et al. [2002] introduced a probabilistic search model for human motion tracking and synthesis from motion capture data. This model structures motion data as a binary tree using PCA dimensionality reduction, and predicts the next frame of motion based on preceding frames. Lee et al. [2002] proposed a vision-based interface that compares silhouettes of the human body with a set of prerendered images of motion fragments to traverse a motion graph in accordance with the input video. Ramanan and Forsyth [2003] developed a system that utilizes a collection of motion capture data that is annotated semiautomatically using a support vector machine so as to synthesize an annotated human motion which matches the input video. These methods are primarily for synthesizing novel maneuver combinations, whereas ours is for capturing a particular style of motion specified in the input video. Based on the ideas of displacement mapping [Bruderin and Williams 1995; Witkin and Popović 1995] and spacetime formulation [Gleicher 1997, 1998; Lee and Shin 1999], we formulate motion reconstruction as a spacetime constraint problem. We also adopt the hierarchical curve fitting scheme in Lee et al. [1997] and Lee and Shin [1999] to solve this problem.

In the original version of this article, we suggested a new scheme for reconstructing (from a single video stream) a highly dynamic motion, such as a shooting motion in soccer, for a full human body consisting of a 40 DOFs (degrees of freedom) required for realistic character animation [Park et al. 2002]. In the current version, we enhance several aspects of this work. First, we try to minimize user interactions in 2D feature tracking and timewarping. Second, we completely reformulate our video-guided motion synthesis problem to estimate not only joint orientations, but also segment proportions. Accordingly, our new formulation includes two types of parameters to optimize: local parameters such as joint orientations, and global parameters such as camera configuration and segment proportions. Third, to verify the effectiveness of our scheme for motion synthesis, we perform extensive experiments with a variety of video clips. Finally, we show the potential of our scheme as a new paradigm for motion capture, that is, capturing motions from videos taken with a monocular camera outside a motion capture studio.

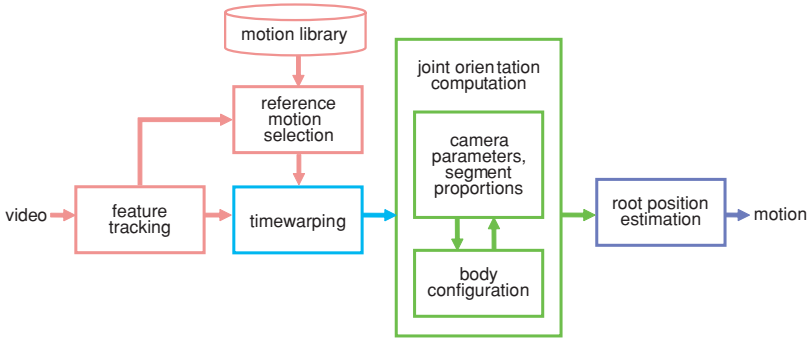


Fig. 1. Block diagram of our synthesis method.

3. OVERVIEW

Our proposed synthesis method consists of four major parts: 2D feature tracking, timewarping, local posture computation, and root position estimation. Figure 1 shows a block diagram that illustrates the overall structure of our method.

In 2D feature tracking, features such as joint positions of the character are obtained frame-by-frame from the input video stream. Since our input video possibly contains highly dynamic motions with unknown camera and character models, the features are tracked semiautomatically. We apply the patch-based segment tracking method of Ju et al. [1996] to the input video stream, and then manually adjust the tracked results when the 2D tracker fails as a result of various causes, such as noises, motion blur, and weak frame coherence.

Using these extracted features, our system selects a reference motion which is similar to the target motion from the library. To do this, we adopt the concept of a view-based indexing method in Ben-Arie et al. [2001a, 2001b]. We extend their approach not only to select a reference motion, but also to obtain a camera configuration.

To synchronize the reference motion with that in the input video, we find the keytimes, that is, the moments of interaction between the character and his or her surrounding environment in both the motion and video. We provide an automatic method for extracting the moments of heel-strikes and toe-offs in the video. Interaction moments in the reference motion are also extracted by adopting the method of Liu and Popović [2002]. Then, we timewarp the reference motion by aligning the keytimes in this motion with corresponding ones in the video. We use the timewarped motion, as an initial guess for the target motion, as well as a guide for optimization.

To obtain the local postures of an articulated figure, that is, joint orientations and segment proportions, 2D features are used as the constraints so that the projected joint positions are coincident with their corresponding features in the video. We select a configuration that minimizes the deviation from the reference motion, while satisfying the constraints. To construct a smooth motion, we adopt a displacement map based on a multilevel B-spline [Lee et al. 1997].

Finally, we estimate a root trajectory to complete the synthesis process. There are two different cases: In the first case, we deal with motion that exhibits some interactions between the character and his or her surrounding environment. To acquire a feasible root trajectory of the target motion that preserves these interactions, we modify the root trajectory of the reference motion while preserving the details of the motion. In the second case, we exploit the dynamic property of the reference motion that should be preserved to construct a root trajectory.

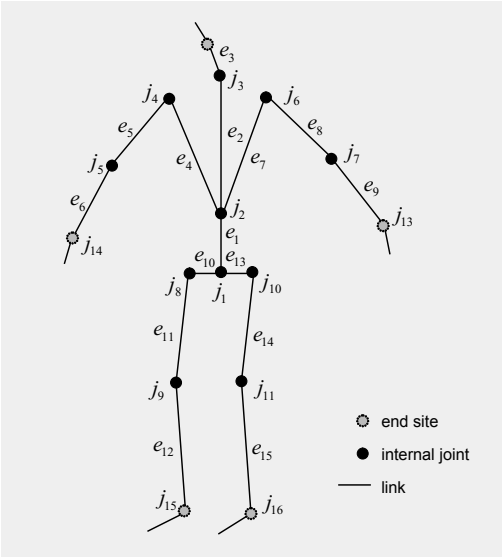


Fig. 2. Articulated figure model.

4. BODY AND CAMERA MODELS

4.1 Body Model

Our articulated figure model consists of n joints formed by m segments. We represent it by a rooted tree (J, E), where $J = \{j_1, \ldots, j_n\}$ and $E = \{e_1, \ldots, e_m\}$ are the set of joints and of segments, respectively. As illustrated in Figure 2, $n = 16$, $m = 15$, and we choose the pelvis joint (j_1) as the root. A joint configuration is given by $(\mathbf{p}_1, \mathbf{q}_1, \ldots, \mathbf{q}_n)^T$, where $\mathbf{p}_1 \in \mathbb{R}^3$ and $\mathbf{q}_1 \in \mathbb{S}^3$ denote the position of the root and its orientation, respectively, and $\mathbf{q}_i \in \mathbb{S}^3$ the orientation of joint i for $2 \leq i \leq n$.

Since we are dealing with the image stream obtained by a single monocular video camera, the absolute length of a segment cannot be inferred from the image stream. Thus, we focus on the segment proportion vector $\mathbf{s} = (s_1, \ldots, s_{m-1})^T = (l_2/l_1, \ldots, l_m/l_1)^T$, rather than the segment lengths themselves, where l_i, $1 \leq i \leq n$ is the length of a segment e_i. Due to reflection symmetry about the sagittal plane of a human body, each segment in one side of this plane has the same proportion as the corresponding segment in the other side. Exploiting this property, we further simplify the segment proportion vector $\mathbf{s}$. As shown in Figure 2, the final segment proportion vector is represented as $\mathbf{s} = (s_1, s_2, s_3, s_4, s_5, s_9, s_{10}, s_{11})^T$, where $(s_6, s_7, s_8, s_{12}, s_{13}, s_{14})^T$ is removed from the original vector.

4.2 Camera Model

To determine the 3D configuration of an articulated figure from its 2D projection onto the image plane, we need to estimate a camera model. We adopt a weak perspective projection model, which is valid when the average variation of the depth of character along the line of sight is small compared to the distance between the camera and character. In many sports videos, this model is a good approximation, since the camera is placed far from the player. In general, a camera model with full degrees of freedom is usually parameterized by 11 parameters (six extrinsic and five intrinsic). We assume that the five intrinsic parameters are ideal; that is, with zero skew and unit aspect ratio (the retina coordinate axes are orthogonal and pixels are square), and that the center of the CCD matrix coincides with the principal point. The only unknown intrinsic parameter is the focal length f.

Together with the six extrinsic parameters, the camera model is represented by $\mathbf{c} = (t_x, t_y, t_z, \theta, \phi, \psi, f)$, where (t_x, t_y, t_z) and (θ, ϕ, ψ) describe the position and orientation of the camera, respectively. We use the tracked root of the articulated figure as the origin of the image plane. This effectively factors out camera translation in a weak perspective model. Combined with the unknown intrinsic parameter f, we reduce (t_x, t_y, t_z) to that of the ratio γ of focal length f to the distance between the root joint of the body and camera. Our reduced camera model is thus parameterized by $\mathbf{c} = (\theta, \phi, \psi, \gamma)$.

For a single monocular image stream, there is inherent ambiguity between the camera and body orientations, since feature positions are determined by their relative relationship. Unfortunately, we cannot resolve this ambiguity with a single image stream, which may result in 3D motion with little frame coherency when the camera parameters are allowed to change from frame-to-frame. With the assumption that the camera model does not change over the frames of a short motion segment, we will find the camera configuration best fitted for the given image sequence. The camera model proposed here is used to compute the joint orientations and body segment proportions of the character in the video. The global root trajectory of the character will be separately estimated later to obtain its global motion.

5. PREPROCESSING

5.1 Feature Tracking

There have been rich research results on feature tracking with or without prior knowledge of human body models. Previous approaches work well under some assumptions, such as constant illumination [Bregler et al. 2002, 2004; Pavlovic et al. 1999], static backgrounds [Gavrila 1999; Wren et al. 1997], and frame coherency [Bregler and Malik 1998; Bregler et al. 2004; Demirdjian et al. 2003; Howe et al. 2000; Morris and Rehg 1998; Sidenbladh et al. 2000b; Pavlovic et al. 1999; Wren et al. 1997]. However, our goal is to synthesize 3D motions from monocular input videos which possibly contain uncalibrated images with cluttered backgrounds. A typical example is a TV broadcasting sports video sequence where an athlete performs highly dynamic motions, which may result in weak frame coherency. Hence, we take a semiautomatic scheme to track the features in the video.

We adopt the method proposed by Ju et al. [1996] for feature tracking, known to be one of the best tracking methods for unknown body and camera models [Gavrila 1999]. Employing this method as a basic tool, we further adjust the tracking results (manually, if needed) to cope with accumulated errors or occlusions. We model each body part, such as limbs and head, as a planar patch whose position and shape are controlled by eight parameters, that is, the four vertex positions of the patch. We determine the parameter values in each frame by minimizing the mismatch between the projection of a body part and the image data. We assume that each patch moves linearly in the image plane, while the brightness of each patch is constant between consecutive frames.

Based on the spacetime constraint formulation in Gleicher [1998] and Lee and Shin [1999], our motion synthesis scheme does not require all feature (joint) positions at every frame. After capturing only unoccluded features, we use them as the spacetime constraints to deform the reference motion while preserving its motion characteristics. Thus, we use only visible features $(\bar{\mathbf{p}}_1(t), \ldots, \bar{\mathbf{p}}_n(t))^T$, where

$$\bar{\mathbf{p}}_i(t) = \begin{cases} \text{the projected position } \mathbf{p}_i(t) \text{ of joint } i, & \text{if it is visible} \\ \emptyset, & \text{if it is not visible.} \end{cases} \tag{1}$$

In our experiments, since input video streams are taken from outdoor scenes such as TV broadcast sports events, 20–30% of the visible features were marked manually due to weak frame coherency and abrupt illumination change.

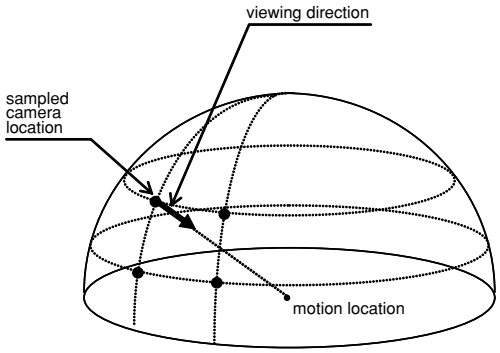


Fig. 3. Sampled camera configurations.

5.2 Reference Motion Selection

In this section, we describe how to select a reference motion that is similar to the target motion specified in a video. Our motion library contains a set of previously captured motion clips. Each clip has a sequence of postures representing a basic motion. Our motion selection scheme is based on the view-based video indexing technique proposed by Ben-Arie et al. [2001a, 2001b]. Their method selects a reference motion similar to the target motion given in the input video by comparing its 2D features with projected features of each motion in the library. To efficiently handle a large motion database, a set of hash tables are used. We extend their method not only to select reference motion, but also to estimate the camera parameters for projection.

To capture a camera configuration, the camera parameter space is sampled regularly, as shown in Figure 3. Based on the camera model described in Section 4.2, we discretize the ranges of azimuth θ and elevation ϕ into 18 and 5 uniform intervals, respectively. The ratio γ of the focal length to the distance between the camera and character is fixed, since we use scale-invariant features to select a reference motion. We also set camera tilt ψ to be zero, since this parameter is near zero in most video sources. The set of discretized camera configurations is denoted by $\{\mathbf{c}_i : 1 \leq i \leq C\}$, where i and C are the index of the discretized camera configuration and total number of discretized configurations, respectively. For efficiency, we discretize the camera configuration space sparsely, since the view-based index scheme is not overly sensitive to variation of viewing direction [Ben-Arie et al. 2001a, 2001b].

A motion is a time-varying function that gives the configuration of an articulated figure. For a figure with n joints, we denote a motion by

$$\mathbf{m}(t) = (\mathbf{p}_1(t), \mathbf{q}_1(t), \ldots, \mathbf{q}_n(t))^T. \tag{2}$$

Let $\mathbf{p}_i^{\mathbf{c}}(t)$, $1 \leq i \leq n$ be the projected position of joint i with camera configuration $\mathbf{c}$. The projection $\mathbf{m}^{\mathbf{c}}(t)$ of a motion $\mathbf{m}(t)$ is described as follows:

$$\begin{aligned} \mathbf{m}^{\mathbf{c}}(t) &= \left(\mathbf{p}_1^{\mathbf{c}}(t), \ldots, \mathbf{p}_n^{\mathbf{c}}(t)\right)^T \\ &= (\mathbf{P}(\mathbf{c})\mathbf{f}_1(\mathbf{m}(t)), \ldots, \mathbf{P}(\mathbf{c})\mathbf{f}_n(\mathbf{m}(t)))^T, \end{aligned} \tag{3}$$

where $\mathbf{P}(\cdot)$ and $\mathbf{f}_i(\cdot)$ are the projection matrix and forward kinematic function for joint i, respectively. The 2D pose at time t in the input video stream is given by a vector

$$\bar{\mathbf{m}}(t) = (\bar{\mathbf{p}}_1(t), \ldots, \bar{\mathbf{p}}_n(t))^T. \tag{4}$$

The vector $\bar{\mathbf{m}}(t)$ specifies the target motion at time t that is to be synthesized (see Section 5.1).

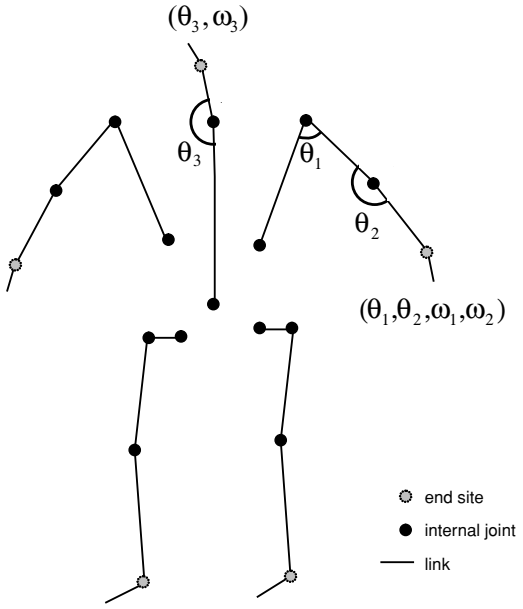


Fig. 4. Decomposed body parts and their corresponding multidimensional tuples.

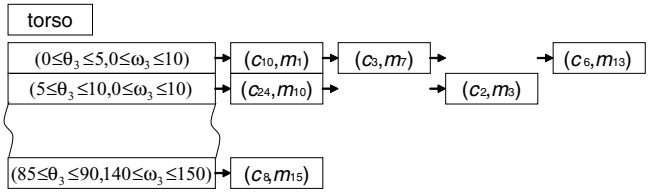


Fig. 5. An example of our hash table.

To ensure scale invariance, Cartesian coordinates $\mathbf{p}_i^{\mathbf{c}}(t), 1 \leq i \leq n$ of the joints are transformed to the projected angles between successive links of an articulated figure. For convenience, we decompose a human-like figure into a set of five body parts: legs, arms, and the torso, as shown in Figure 4. To represent a limb, we use a tuple $(\theta_1, \theta_2, \omega_1, \omega_2)$, where θ_1 and θ_2 denote the angle between the forearm (calf) and upper arm (thigh) and that between the upper arm (thigh) and chest (pelvis), respectively. Here, ω_1 and ω_2 represent the corresponding angular velocities. For the torso, we use a tuple (θ_3, ω_3), where θ_3 and ω_3 represent the angle between the pelvis and chest and its corresponding angular velocity, respectively.

For each 3D motion in the library, we compute its projections $\mathbf{m}^{\mathbf{c}_i}(t)$ for different camera configurations $\mathbf{c}_i, 1 \leq i \leq C$. Then, each projection $\mathbf{m}^{\mathbf{c}_i}(t)$ at every frame t is decomposed into five body parts, as described earlier. The tuple of each body part is quantized to provide an entry for its corresponding hash table. We have five hash tables: one for the torso and four for the limbs (one per limb). It is important to create a different hash table for each body part. This facilitates reference motion selection only with unoccluded body parts. In particular, we index a motion by using only visible body parts. As illustrated in Figure 5, each entry of a hash table contains a list of an ordered pair of indices that represent a camera configuration and a 3D motion, respectively.

Our remaining task is to select a reference motion using the hash tables constructed for motions in the library, provided with the input video. To do this, we choose a set of representative frames from the input video. We prefer frames wherein interactions with the environment occur. If there are no such interactions, we sample a number of frames (6–7 frames in our experiments) at random, while keeping the time between adjacent samples greater than a given threshold.

Given the set of representative frames in the video, the 2D pose at each of these frames is disassembled into five parts, and quantized to index their corresponding hash tables. Let $v_k(i,m)$ be the sum of votes for motion clip m and camera configuration $\mathbf{c}_i$ by unoccluded body parts at representative frame k. We add $v_k(i,m)$ together over all representative frames k to form a vote array $\{\sum_{k=1}^{K} v_k(i,m) : 1 \le i \le C\}$ for every motion m.

With this vote array, we choose the maximum vote $V(m)$ for each motion m over all camera configurations:

$$V(m) = \max_i \left\{ \sum_{k=1}^{K} v_k(i,m) \right\}, \qquad 1 \le m \le M \tag{5}$$

This also gives the camera configuration for motion m. Finally, we choose as a reference motion $\mathbf{m}^r$ the motion which maximizes $V(m)$, that is,

$$\mathbf{m}^r = \arg\max_m \{V(m)\}. \tag{6}$$

5.3 Interaction Moment Detection

Given the reference motion $\mathbf{m}^r(t)$ and 2D images $\bar{\mathbf{m}}(t)$ in the video, we establish a time correspondence between $\mathbf{m}^r(t)$ and $\bar{\mathbf{m}}(t)$. It is well-known that the dynamic timewarping technique gives optimal sample correspondences between two functions [Bruderin and Williams 1995; Demori and Probst 1986]. However, the camera estimated in Section 5.2 is a rough approximation of the actual camera used for the video, and some input features may be missing due to occlusion. Moreover, the dimensions of the 3D reference motion $\mathbf{m}^r(t)$ and 2D input image stream $\bar{\mathbf{m}}(t)$ are different. Thus, this technique cannot be directly applied in our case.

To address these issues, we start with a set of keytimes in the video, that is, the moments of interaction between the character and his or her surrounding environment. For example, we choose instances of heel-strikes and toe-offs as the keytimes for human locomotion. Detecting these instances in 3D motion is well-known [Liu and Popović 2002]. However, detecting such instances in a video is rather challenging for the following reasons: First, the camera may move along with the characters. Second, the geometry of the scene is not always available. In general, it is hard to discriminate the motion of a camera from that of an object in a monocular video [Gavrila 1999].

In this article, we propose a practical solution to attack this problem. Consider a kicking motion of a soccer player, as shown in Figure 6. To keep the character at the center of the image plane, we assume that the camera moves along a line. This assumption is reasonable for a short video clip, since our weak perspective camera model allows only translations to track the character. As shown in Figure 6(a), we define a new coordinate frame in the image plane with reference to this line, which can be found by applying principal component analysis to foot positions in the video. We apply our scheme to left and right foot positions separately to detect the interaction moments for both feet.

Let $\{p_e(i), 1 \le i \le n\}$ be the set of left (or right) foot positions in the image plane, where n is the number of frames. We apply principal component analysis to this set to find the principal axes, as shown in Figure 6(a). In general, the variation of foot positions is maximal along the moving direction of the character, since footstep length is larger than ankle height during locomotive motions. The $\hat{x}$ axis in Figure 6 corresponds to the direction with maximum projected variation. The camera moves linearly

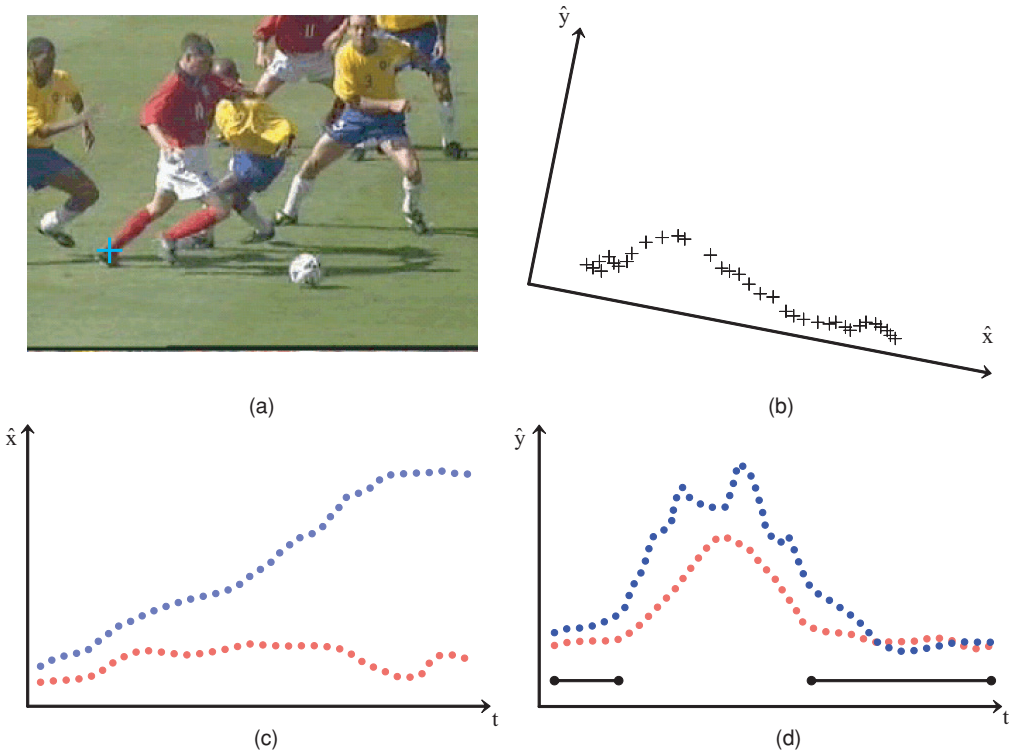


Fig. 6. (a) Finding interaction moments in the video; (b) principal axes; (c) position (blue curve) and velocity (red curve) along the $\hat{x}$ axis; (d) position (blue curve) and velocity (red curve) along the $\hat{y}$ axis.

above a plane parallel to the ground, and the character also moves linearly on the same ground plane. Therefore, this axis indicates the direction of relative movement of the character with respect to the camera, which is projected onto the image plane. The $\hat{y}$ axis corresponds to the direction perpendicular to the $\hat{x}$ axis along which projected foot positions show the minimum variation. Thus, the $\hat{y}$ axis is not necessarily coincident with the projected global-up direction. The variation of data along this direction in the image plane mainly originates from the foot movement perpendicular to the moving direction of the character. Let $\hat{p}_e(i)$, $1 \leq i \leq n$ be the foot trajectory in the $\hat{x}$–$\hat{y}$ coordinate frame. The time-varying position and velocity of the $\hat{x}$-component of $\hat{p}_e(i)$ are shown in Figure 6(b). Figure 6(c) illustrates the corresponding position and velocity of the $\hat{y}$-component. While the foot is in contact with the ground, the variation of foot positions and velocities along the $\hat{y}$-coordinate are less than the threshold values for some consecutive frames, as shown in Figure 6(c). Thus, interaction moments such as heel-strikes and toe-offs are the start and end frames of such consecutive frames, respectively.

Our scheme works well for finding the interaction moments of a player in TV broadcast sports video sequences. However, our method is designed mainly for detecting heel-strikes and toe-offs for locomotive motions. We manually mark other kinds of interactions, such as moments of kicking a ball.

6. PROPORTION AND ORIENTATION COMPUTATION

6.1 Timewarping

The objective of this stage is to establish a time correspondence between the reference motion $\mathbf{m}^r(t)$ and its corresponding input images $\bar{\mathbf{m}}(t)$. First, using the keytimes extracted in Section 5.3, we align the

reference motion with the input video to identify the portion of reference motion to be used for motion synthesis. We assume that the reference motion is long enough to cover the input video stream. For a cyclic motion, we concatenate the reference motion so as to cover the input video.

First, we encode every keytime t_i in both the video and reference motion into a symbol $S(t_i)$ as follows:

$$S(t_i) = \begin{cases} s_1, \text{ if } t_i \text{ is a moment of left heel-strike} \\ s_2, \text{ if } t_i \text{ is a moment of left toe-off} \\ s_3, \text{ if } t_i \text{ is a moment of right heel-strike} \\ s_4, \text{ if } t_i \text{ is a moment of right toe-off} \\ s_5, \text{ if } t_i \text{ is a moment of user-specified keytimes,} \end{cases} \tag{7}$$

where t_i is the i-th keytime extracted in Section 5.3 and $s_i, 1 \leq i \leq 5$ are symbols. Then, we find the segment of the reference motion that is best matched with the input video by employing a string matching algorithm [Cormen et al. 1999]. If there exist two or more matched segments, we choose the one whose length is most similar to that of the motion in the video.

In addition to the keytimes for interaction moments, the start and end frames should be specified for timewarping. We describe how we choose the start frame. If the first frame of the video has a keytime, then we choose as the start frame that with the first keytime of the reference motion. Otherwise, we first determine where to search for the start frame in the reference motion. Let d be the number of frames from the first frame to the first keytime of the video. We scale d by the ratio of number of frames in the string-matched segment of the reference motion to that in the video, and subtract this from the first keytime of the reference motion to locate the first guess of the start frame. In the (user-specified) vicinity of this frame, we choose as the start frame that whose projected pose is most similar to the start frame of the video. The end frame can be chosen in the symmetrical manner.

Our remaining task is to timewarp the reference motion $\mathbf{m}^r(t)$ to align its keytime sequence with that of the input video $\bar{\mathbf{m}}(t)$. Let $K = \{t_1, \ldots, t_c\}$ be a set of keytimes for the reference motion and $\bar{K} = \{\bar{t}_1, \ldots, \bar{t}_c\}$ the counterpart for the video stream. To make K coincide with $\bar{K}$, we use a piecewise linear warping function:

$$\bar{t} = \bar{t}_k + \left(\frac{\bar{t}_{k+1} - \bar{t}_k}{t_{k+1} - t_k} \right) (t - t_k) \ , \tag{8}$$

where $t_k \leq t \leq t_{k+1}$.

6.2 Formulation

6.2.1 *Kinematic Constraints.* The projected joint positions of the articulated figure need to coincide with their corresponding features at each frame of the video stream. The input video is taken from an uncalibrated camera with an unknown trajectory, and reference objects are not always available in the video stream. Therefore, we describe the joint positions relatively to the root segment. In other words, the configuration of the articulated figure is $\mathbf{x}(t) = \mathbf{m}(t)|_{\mathbf{p}_1(t)=\mathbf{0}_3}$, that is, $\mathbf{x}(t) = (\mathbf{0}_3, \mathbf{q}_1(t), \ldots, \mathbf{q}_n(t))^T$, where $\mathbf{0}_3$ is a zero vector.

Provided with the vector $\mathbf{s}$ of segment proportions, camera configuration $\mathbf{c}$, and input 2D position $\bar{\mathbf{p}}_i(t)$ of joint i at time t, the kinematic constraint for joint i of the articulated figure at time t is given as follows:

$$\|\bar{\mathbf{p}}_i(t) - \mathbf{P}(\mathbf{c})\mathbf{g}_i(\mathbf{s}, \mathbf{x}(t))\| = 0, \tag{9}$$

where $\mathbf{g}_i(\cdot, \cdot)$ and $\mathbf{P}(\cdot)$ are the forward kinematic function for joint i and the projection matrix, respectively. More precisely, Eq. (9) gives the constraint for a projected joint position. For our convenience,

however, we call this a kinematic constraint in order to imply that the constraint originates from a kinematic configuration.

6.2.2 *Objective Function.* Due to depth ambiguity, there can be multiple configurations that satisfy the kinematic constraints given by Eq. (9). DiFranco et al. [1999] pointed out that the depth ambiguity can be resolved partially using some additional constraints, such as joint angle limits. Even with such additional constraints, however, the motion synthesis problem is often still underconstrained due to the excessive degrees of freedom of an articulated figure. Furthermore, the forward kinematic function $\mathbf{g}_i(\cdot,\cdot)$ depends on not only the joint configuration but also on segment proportions. That is, a change of segment proportions results in different projected joint positions for the same joint configuration. Therefore, we consider the segment proportions and joint configuration simultaneously. To achieve the best configuration and segment proportions, we exploit the reference motion as well as anthropometrical segment proportion data.

We model the distribution of segment proportions using the anthropometric measurements given in Pheasant [1996]. The segment proportion vector has multivariate normal distribution:

$$p(\mathbf{s}) = ((2\pi)^k |\Sigma|)^{-1/2} \exp(-1/2(\mathbf{s}-\bar{\mathbf{s}})^T \Sigma^{-1}(\mathbf{s}-\bar{\mathbf{s}})), \tag{10}$$

where $\mathbf{s}$ is a random vector that represents the segment proportions, k is the number of elements in $\mathbf{s}$ (in our case $k = 8$), and $\bar{\mathbf{s}}$ and Σ are the mean and covariance matrix of the segment proportions, respectively. We minimize the deviation of segment proportion vector $\mathbf{s}$ from mean $\bar{\mathbf{s}}$, that is,

$$\text{deviation}(\mathbf{s}, \bar{\mathbf{s}}) = (p(\mathbf{s}) - p(\bar{\mathbf{s}}))^2, \tag{11}$$

to obtain an anthropometrically plausible human model.

Given the segment proportions, we now determine the joint configuration. The joint configuration $\mathbf{x}$ should be as similar as possible to reference motion $\mathbf{x}^r$ in order to synthesize a convincing motion. The difference between $\mathbf{x}$ and $\mathbf{x}^r$ over all t is defined as follows:

$$\int_t \text{dist}(\mathbf{x}^r(t), \mathbf{x}(t))dt = \int_t \sum_{i=1}^{n} \left\| \ln((\mathbf{q}_i(t))^{-1}\mathbf{q}_i^r(t)) \right\|^2 dt, \tag{12}$$

where $\ln(\cdot)$ is the logarithmic map of unit quaternions [Shoemake 1985]. Therefore, we find a configuration $\mathbf{x}$ and segment proportion vector $\mathbf{s}$ by minimizing the following objective function:

$$g(\mathbf{x}, \mathbf{s}) = (p(\mathbf{s}) - p(\bar{\mathbf{s}}))^2 + \omega \int_t \text{dist}(\mathbf{x}^r(t), \mathbf{x}(t))dt. \tag{13}$$

Here, ω is a weighting factor combining different measures. The first term describes the deviation of segment proportion vector $\mathbf{s}$ from mean $\bar{\mathbf{s}}$, and the second measures the deviation of the configuration of the figure from that of the reference motion over all frames.

6.3 Numerical Optimization

6.3.1 *Basic Idea.* Our motion synthesis problem can be reduced to that of finding the body configuration $\mathbf{x}$, camera configuration $\mathbf{c}$, and segment proportion vector $\mathbf{s}$ that minimize the objective function, while satisfying the constraints as formulated in the previous section. We transform the constrained optimization problem into an unconstrained version to obtain a new objective function:

$$\begin{aligned} \hat{g}(\mathbf{x}, \mathbf{s}, \mathbf{c}) \;=\; & \int_t \sum_{i=1}^{n} \|\bar{\mathbf{p}}_i(t) - \mathbf{P}(\mathbf{c})\mathbf{g}_i(\mathbf{s}, \mathbf{x}(t))\|^2 dt \\ & + \omega_1(p(\mathbf{s}) - p(\bar{\mathbf{s}}))^2 + \omega_2 \int_t \text{dist}(\mathbf{x}^r(t), \mathbf{x}(t))dt, \end{aligned} \tag{14}$$

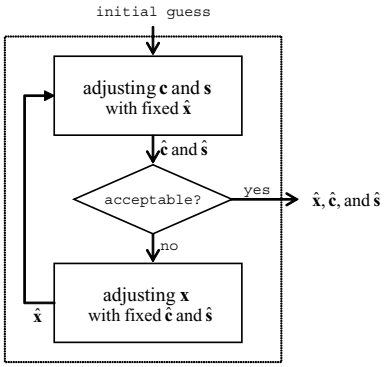


Fig. 7. Iterative scheme to solve Eq. (14).

where ω is a weighting factor. The first term on the righthand-side of Eq. (14) reflects kinematic constraints given in Eq. (9), and the second and third terms are from the original objective function described in Eq. (13). In the new formulation, kinematic constraints do not paly the role of hard constraints any longer. However, our formulation tries to minimize the error to satisfy these constraints.

Eq. (14) has two different sets of parameters (unknowns): the set of local parameters and that of global parameters. The former is for the joint configuration, and the latter for both the camera configuration and segment proportions. We may solve the equation for all the parameters at once. However, it would not give good convergency. Thus, we alternately optimize one set of parameters while fixing the other, as illustrated in Figure 7. In the following sections, we describe each step of this iterative process.

In numerical optimization, a good initial guess is important to obtain a good solution [Gill and Murray 1974; Fletcher 1980; Press et al. 1992]. We use the timewarped reference motion as the initial estimate of the target configuration. The initial camera parameters are estimated in the reference motion selection stage (see Section 5.2). The initial segment proportions are selected as the mean of the anthropometric distribution given by Eq. (10). Moreover, the reference motion and segment distribution function are good guides for the two alternating steps, respectively.

6.3.2 *Computing Camera Parameters and Segment Proportions.* Provided with a fixed joint configuration $\hat{\mathbf{x}}$, this step is to compute both the camera configuration and segment proportions. Thus, the objective function is simplified, as follows:

$$\sum_{k=1}^{m}\sum_{i=1}^{n} \|\bar{\mathbf{p}}_i(k) - \mathbf{P}(\mathbf{c})\mathbf{g}_i(\mathbf{s}, \hat{\mathbf{x}}(k))\|^2 + \omega_1(p(\mathbf{s}) - p(\bar{\mathbf{s}}))^2 + c_1, \tag{15}$$

where m is the number of input frames. Here, the last term of the righthand-side of Eq. (14) is reduced to a constant c_1, since the joint configuration is fixed. We adopt the conjugate gradient method [Press et al. 1992] to minimize this objective function.

6.3.3 *Computing Joint Configuration.* In this step, we obtain a joint configuration, given the camera configuration and segment proportions. The joint configuration $\mathbf{x}$ consists of a bundle of signals that describe joint orientations. These signals are sampled at every discrete time instance (frame). At each constrained frame, we optimize the joint configuration independently to satisfy the kinematic constraints at this frame. However, this may cause undesirable jerkiness, since interframe coherence is not considered. Such jerkiness can also be caused by noises in the input features. We employ the multilevel B-spline fitting technique [Lee et al. 1997; Lee and Shin 1999] to take into account the interframe coherence as well.

Inverse Kinematics Solver. We discretize Eq. (14) to obtain the new objective function:

$$\sum_{k=1}^{m}\left\{\sum_{i=1}^{n}\|\bar{\mathbf{p}}_i(k)-\mathbf{P}(\hat{\mathbf{c}})\mathbf{g}_i(\hat{\mathbf{s}},\mathbf{x}(k))\|^2+\omega_2\text{dist}(\mathbf{x}^r(k),\mathbf{x}(k))\right\}+c_2, \tag{16}$$

where $\hat{\mathbf{c}}$ and $\hat{\mathbf{s}}$ are the fixed camera configuration and segment proportion vector, respectively. The second term of the righthand-side of Eq. (14) is simplified to a constant c_2, since the segment proportions are fixed. At each constrained frame k, we solve for the joint configuration $\mathbf{x}(k)$ that minimizes the sum of errors caused by kinematic constraints at this frame, that is, we minimize

$$\sum_{i=1}^{n}\|\bar{\mathbf{p}}_i(k)-\mathbf{P}(\hat{\mathbf{c}})\mathbf{g}_i(\hat{\mathbf{s}},\mathbf{x}(k))\|^2+\omega_2 dist(\mathbf{x}^r(k),\mathbf{x}(k)). \tag{17}$$

Again, we employ the conjugate gradient method [Press et al. 1992].

Interframe Coherence. Numerical optimization produces joint configurations at a subset of frames where constraints are specified. These configurations are used to compute the displacement from corresponding configurations in the reference motion. Then, the joint configurations of the remaining frames can be produced from these displacements. This process consists of interpolating the displacements with a multilevel B-spline, and deforming the reference motion on the remaining frames by adding the new displacements appropriately. To do this, we combine motion displacement mapping [Bruderin and Williams 1995; Witkin and Popović 1995] with multilevel B-spline fitting [Lee et al. 1997; Lee and Shin 1999].

A motion displacement map describes the difference between two motions. In our case, the displacement map $\mathbf{d}$ for reference motion $\mathbf{x}^r$ and target motion $\mathbf{x}$ is defined as follows:

$$\begin{aligned}\mathbf{d}&=\mathbf{x}\ominus\mathbf{x}^r\\&=(\mathbf{0}_3,\mathbf{q}_1,\ldots,\mathbf{q}_n)^T\ominus\left(\mathbf{0}_3,\mathbf{q}_1^r,\ldots,\mathbf{q}_n^r\right)^T\\&=\left(\mathbf{0}_3,\ln\left(\left(\mathbf{q}_1^r\right)^{-1}\mathbf{q}_1\right),\ldots,\ln\left(\left(\mathbf{q}_n^r\right)^{-1}\mathbf{q}_n\right)^T,\\&=(\mathbf{0}_3,\mathbf{v}_1,\ldots,\mathbf{v}_n)^T\end{aligned} \tag{18}$$

where $\mathbf{v}_i\in\mathbb{R}^3$ is the rotation vector of joint j_i for $1\le i\le n$. Thus, a new motion can be obtained by adding the displacement map to the original motion, that is,

$$\begin{aligned}\mathbf{x}&=\mathbf{x}^r\oplus\mathbf{d}\\&=\left(\mathbf{0}_3,\mathbf{q}_1^r,\ldots,\mathbf{q}_n^r\right)^T\oplus(\mathbf{0}_3,\mathbf{v}_1,\ldots,\mathbf{v}_n)^T\\&=\left(\mathbf{0}_3,\mathbf{q}_1^r\exp(\mathbf{v}_1),\ldots,\mathbf{q}_n^r\exp(\mathbf{v}_n)\right)^T.\end{aligned} \tag{19}$$

Given the displacement vector $\mathbf{d}(i)$ at each constrained frame i, we compute a smooth displacement map $\mathbf{d}(t)$ that interpolates $\mathbf{d}(i)$ for all i within a given tolerance. We employ the multilevel B-spline approximation technique [Lee et al. 1997; Lee and Shin 1999], which uses a series of B-spline functions with different knot spacings on the same interval. In contrast to local curve fitting with B-splines, the hierarchical structure of multilevel B-spline fitting can make a smooth shape, without undulations, by globally propagating errors at coarse levels and adding details at fine levels. Finally, we apply the smooth displacement map $\mathbf{d}(t)$ to the reference configuration $\mathbf{x}^r(t)$ so as to achieve the final configuration $\mathbf{x}(t)$, as follows:

$$\mathbf{x}(t)=\mathbf{x}^r(t)\oplus\mathbf{d}(t). \tag{20}$$

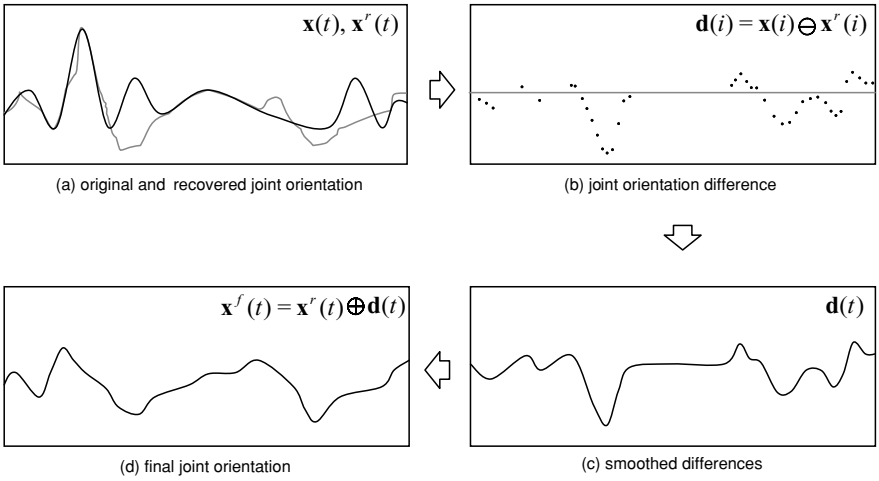


Fig. 8. (a) The curves represent one component of a unit quaternion of the reference motion and that of the recovered motion, respectively; (b) joint orientation displacement vectors $\mathbf{d}(i)$ at each constrained frame; (c) displacement map $\mathbf{d}(t)$ that approximates the displacement vectors; (d) sum of the reference motion and displacement map.

This procedure is illustrated in Figure 8. Here, we need to trade off the smoothness of recovered joint orientations against their approximation errors, depending on the quality of the 2D features. By properly choosing the resolution of knots for fitting, we can synthesize a smooth motion, even with noisy input data, while maintaining acceptable accuracy.

7. ROOT POSITION ESTIMATION

In the previous sections, we have described a method to obtain the joint orientations of an articulated figure. Now, we describe how to construct a proper trajectory of the root segment to synthesize a full 3D motion. This motion $\mathbf{m}(t)$ is the direct sum of the joint configuration $\mathbf{x}(t)$ and displacement map $\mathbf{d}(t)$ that describes only the translational movement of the root segment:

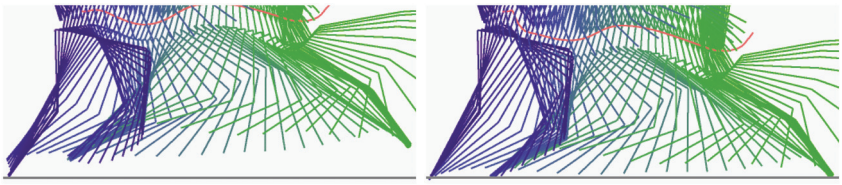
$$\begin{aligned}\mathbf{m} = \mathbf{x} \oplus \mathbf{d} &= (\mathbf{0}_3, \mathbf{q}_1, \ldots, \mathbf{q}_n)^T \oplus (\mathbf{p}_1, \mathbf{0}_3, \ldots, \mathbf{0}_3)^T \\ &= (\mathbf{p}_1, \mathbf{q}_1, \ldots, \mathbf{q}_n)^T,\end{aligned} \tag{21}$$

where $\mathbf{p}_1(t)$ is the root trajectory in the global frame. Since the camera may move along with the characters, the actual trajectory of the character is hard to synthesize with only the information given in the video. We try to construct a plausible root trajectory while satisfying the kinematic constraints and dynamic property of the reference motion.

We discriminate between two classes of motions (according to their interaction with the environment) which are the sources of constraints. In the first case, we deal with motion that exhibits some interactions between the character and the surrounding environment. Locomotion is a typical example, since the feet of the character contact the ground. In the second case, we deal with motion that does not show such interactions; a jumping motion is a typical example. In each case, we describe how to obtain the displacement map $\mathbf{d}(t)$, particularly, the root trajectory $\mathbf{p}_1(t)$.

7.1 Motion Involving Interactions With the Static Environment (Case 1)

Consider the kicking motion of a soccer player, as shown in Figure 9. The synthesized motion is so dynamic that the root trajectory is quite different in height from that of the reference motion. Therefore,



(a) using the reference root trajectory (b) using the adjusted trajectory

Fig. 9. Estimating a root trajectory of a motion involving interaction.

we adjust the height of the root segment to make the stance foot come in contact with the ground at every constrained frame. We start with the motion $\hat{\mathbf{m}}(t) = \mathbf{x}(t) \oplus (\mathbf{p}_1^r(t), \mathbf{0}_3, \ldots, \mathbf{0}_3)^T$, where $\mathbf{x}(t)$ and $\mathbf{p}_1^r(t)$ are the synthesized joint configuration, as explained in Section 6, and root trajectory of the timewarped reference motion, respectively. We compute the displacement $\mathbf{d}(i)$ between the stance foot in the motion $\hat{\mathbf{m}}(t)$ and the ground at every constrained frame i, and then construct a smooth displacement map $\mathbf{d}(t)$ that approximates $\mathbf{d}(i)$ using a multilevel B-spline approximation method. With $\hat{\mathbf{m}}(t) \oplus (\mathbf{d}(t), \mathbf{0}_3, \ldots, \mathbf{0}_3)^T$ as the initial guess, we adopt the motion retargeting technique [Lee and Shin 1999] to determine the final target motion $\mathbf{m}(t)$.

7.2 Motion Without Involving Interactions (Case 2)

Unlike the previous case, the motion in this case does not involve any interaction with the environment, as observed in a jump motion. We exploit the reference motion to determine the root trajectory. Given an initial velocity with no external forces except for gravity, the center of gravity (COG) of an object follows a parabolic trajectory. The COG trajectory $\mathbf{cog}^r(t)$ of the reference motion is defined as follows:

$$\mathbf{cog}^r(t) = \mathbf{p}_1^r(t) + \frac{\sum_{i=1}^n m_i \tilde{\mathbf{p}}_i^r(t)}{\sum_{i=1}^n m_i}, \tag{22}$$

where $\tilde{\mathbf{p}}_i^r(t)$ and m_i for $1 \le i \le n$ represent the vector from the root position $\mathbf{p}_1^r(t)$ to the COG of segment i and its corresponding mass, respectively. Since the reference motion is timewarped linearly, as described in Section 6.1, we also linearly scale the COG trajectory of the reference motion to approximate the COG trajectory $\mathbf{cog}(t)$ of the target motion, as follows:

$$\mathbf{cog}(t) = \mathbf{p}_1(t) + \frac{\sum_{i=1}^n m_i \tilde{\mathbf{p}}_i(t)}{\sum_{i=1}^n m_i} = \eta\, \mathbf{cog}^r(t), \tag{23}$$

where η denotes the scaling factor for timewarping. Here, $\tilde{\mathbf{p}}_i(t)$ can be obtained from the synthesized joint configuration $\mathbf{x}(t)$. Thus, the root trajectory $\mathbf{p}_1(t)$ is computed by combining Eqs. (22) and (23):

$$\mathbf{p}_1(t) = \eta \mathbf{p}_1^r(t) + \left(\frac{\sum_{i=1}^n m_i (\eta \tilde{\mathbf{p}}_i^r(t) - \tilde{\mathbf{p}}_i(t))}{\sum_{i=1}^n m_i} \right). \tag{24}$$

As shown in Figure 10, synthesized motion with the root trajectory of the reference motion follows an infeasible, distorted COG trajectory. However, that with the COG trajectory of the reference motion follows a smooth parabolic trajectory.

8. EXPERIMENTAL RESULTS

We used a human model of 40 DOFs for body configuration (6 DOFs for the pelvis position and orientation, 3 for the chest, 3 for the neck, and 7 for each limb) and 8 DOFs for body segment proportions (see

(a) using the reference root trajectory (b) using the reference COG trajectory

Fig. 10. Estimating a root trajectory during a jump motion. The curves represent the COG trajectories of synthesized motions.

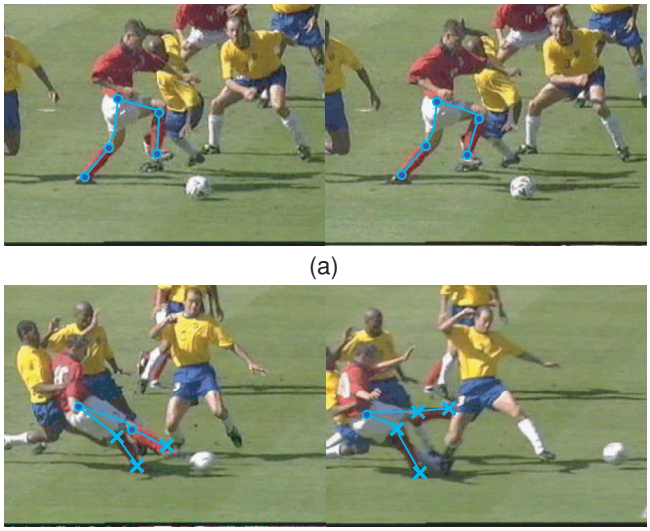


Fig. 11. 2D tracking and interactive adjustment: The circled points are tracked automatically, and the crossed points are manually marked.

Figure 2 for our articulated figure model). In our experiments we used soccer motions, including kicking, running, and header motions, which are highly dynamic. The motion clips were sampled at a rate of 60 frames per second. The video clips were from TV broadcast sports programs. The sampling rate of video clips was 29.97 frames per second, which is a standard NTSC format. The video clips mainly contained "sideway" motions, where a figure is moving across the image (and not pointed directly at the camera), since this camera configuration effectively captures the movement of each segment in locomotive motion. Our synthesis method was implemented in C++ on top of MS Windows XP™ and the TGS OpenInventor™, which is a convenient toolkit to support 3D primitives. Experiments were performed on a Pentium PC (Intel PentiumIV 2.4GHz processor and 1GB *memory*).

8.1 Preprocessing

In this section, we performed our experiments for 2D feature tracking, reference motion selection, and interaction moment detection. The first experiment was for 2D feature tracking. As shown in Figure 11(a), the patch-based 2D feature tracking method works well, in general, under constant

Table I. Motion Group Selection: Average Votes (Standard Deviations) for Motion Groups

	# of Motions	Kick	Header	Walk	Run	Overhead
kick	18	**30.21 (3.24)**	9.24 (1.30)	13.21 (0.74)	11.1 (1.00)	11.98 (2.10)
header	15	8.82 (1.74)	**31.54 (2.98)**	10.34 (1.23)	14.21 (0.97)	10.21 (0.98)
walk	3	13.32 (2.43)	10.11 (1.21)	**26.32 (1.87)**	18.21 (3.21)	7.92 (2.32)
run	3	10.21 (3.21)	11.54 (2.31)	18.98 (3.27)	**27.32 (3.94)**	6.32 (1.99)
overhead	1	7.67 (0.68)	12.80 (0.86)	9.57 (1.04)	8.72 (1.00)	**33.0 (0.00)**

The best score in each row appears in bold face.

illumination when the feature differences between consecutive frames are small. However, for outdoor scenes, the 2D tracker does not work well at times due to low contrast, occlusion, and illumination changes. Figure 11(b) shows an example of tracking failure. The positions of the left ankle and knee were not tracked properly, since the right leg had cast a shadow on the left leg. Those of the right ankle and knee were not tracked well either, since the temporal coherence was too weak to assume constant brightness and motion linearity [Ju et al. 1996]. In this case, we manually marked the 2D features. The features both automatically tracked and manually marked were used as the spacetime constraints to deform the reference motion while preserving its motion characteristics.

The second experiment was for reference motion selection. The repertoire of our motion library consists of 40 motion clips, including running, kicking, and header motions, which are basic soccer motions. They are manually classified into five groups, according to their similarity: kick, header, walk, run, and overhead kick. For example, "instep kick," "inside kick," and "outside kick" are categorized into a group called the "kick" motion group. We applied our reference motion selection scheme to these motions. We first showed how well our scheme can select a motion group. Each of the projected motions in the library was compared to every 3D motion in the same library to vote for the most similar motion. For projections, we used our weak perspective camera model, described in Section 4.2. The camera azimuth θ and elevation ϕ were derived from the average of normals of the character's sagittal plane over all frames, and the camera tilt ψ was set to zero. We set the ratio γ as fixed, since our motion selection scheme is scale-invariant. Then, we accumulated the votes for motions in the same group for every input motion group. Table I shows the average number of votes for the motions in each group. As described in Section 5.2, the votes at every frame were added up and divided by the number of motions in each group. Each row corresponds to the projected motions in each group, while a column corresponds to a 3D motion group in the library. The votes along the diagonal were the highest in each row, which indicates that our scheme chose a proper motion group. For example, if the input image shows a kick motion, our system correctly chooses the kick motion group.

Given a motion group, we next measured how accurately our scheme can distinguish each motion in the same group. Our motion selection scheme was applied to five different motions in the kick motion group. In this experiment, every motion in the group is projected under ten different camera configurations, sampled at random, to prepare a sequence of synthetic input images. Table II shows the average number of votes for each motion in the kick motion group. Again, the votes for 3D motions along the diagonal were sufficiently large, demonstrating that our motion selection scheme always chose the proper reference motion.

The last experiment was to measure how accurately our method can detect the interaction moments in the video. In this experiment, we applied our scheme to TV broadcast sports video sequences. We used

Table II. Reference Motion Selection: Average Votes (Standard Deviations) for Motions in the Same Group

	Inside	Instep	Outside	Toe	Hill
inside	**34.67(4.31)**	22.41(3.22)	23.31(2.31)	18.10(1.08)	11.98(2.10)
instep	17.74(2.63)	**36.45(3.12)**	20.63(1.87)	16.89(1.79)	20.86(1.11)
outside	16.78(3.18)	20.21(3.54)	**34.21(4.42)**	19.01(2.30)	13.18(2.43)
toe	22.35(4.14)	18.87(3.18)	16.98(2.72)	**29.23(4.05)**	17.88(2.71)
hill	19.21(2.86)	21.02(1.63)	17.57(2.11)	21.27(3.34)	**32.83(3.44)**

Table III. Interaction Moment Detection for TV Broadcast Video Sequences

	Video1	Video2	Video3	Video4	Video5	Video6
# of actual keytimes	5	5	6	6	4	0
# of detected keytimes	5	5	6	6	4	0
average frame difference	1.3	1.45	1.33	1.83	1.5	0
maximum frame difference	2	3	2	3	2	0

six video clips for kicking, running, and header motions. Since the actual interaction moments in these video clips were not available, we manually marked the interaction moments, which were regarded as ground-truth values. As shown in Table III, our scheme did not miss any foot interaction moments for these test videos. The interaction moments detected by our scheme were close to the manually marked moments. The maximum frame difference was less than three frames for all input videos.

We also applied our scheme to an existing motion of an articulated figure, that is, a sequence of locomotive motions (3,347 frames). This motion has 96 interaction moments such as heel-strikes and toe-offs. We employed the method suggested in Liu and Popović [2002] to detect these interaction moments. We obtained the input videos by projecting the 3D motion under two different camera configurations, which were chosen at random. Figure 12 shows the results of our interaction moment detection scheme. The camera is allowed to move on the ground plane, while keeping the character's pelvis at the center of the image plane. The horizontal axis describes the index of interaction moments, and the vertical axis describes the time (frame) difference between detected interaction moments in 3D motion and their corresponding 2D input video streams. For both camera configurations, the errors were distributed within three frames, and their mean and variance were 0.5787(0.5138) and 0.6057(0.6924), respectively.

8.2 Motion Synthesis

In motion synthesis, we performed four experiments. The first was to show the accuracy of our scheme for the estimation of joint orientations and segment proportions with synthetic images. The second was for ground-truth comparison using motions acquired in a motion capture studio. The third was to demonstrate the effectiveness of our scheme for motion synthesis, and the final experiment was to show the potential of our scheme as a new paradigm for motion capture.

In the first experiment, we applied our scheme to sequences of synthetic images acquired from an existing motion of an articulated figure (base model). To obtain these image sequences, we retargeted this motion to three body models with different segment proportions (see Figure 13). The model in the middle has segment proportions coincident with the mean anthropometric data given in Pheasant [1996] (see Section 4). The left (respectively, right) model has relatively shorter legs and longer arms (respectively, longer legs and shorter arms) than one in the middle. For each of the three models, we estimated joint orientations and segment proportions using our motion synthesis scheme. Table IV shows the estimated data for selected segments and joints. The deviations of estimated joint orientations were within three degrees of the actual motion for all frames, and those of estimated segment proportions were within three percent of the actual proportions.

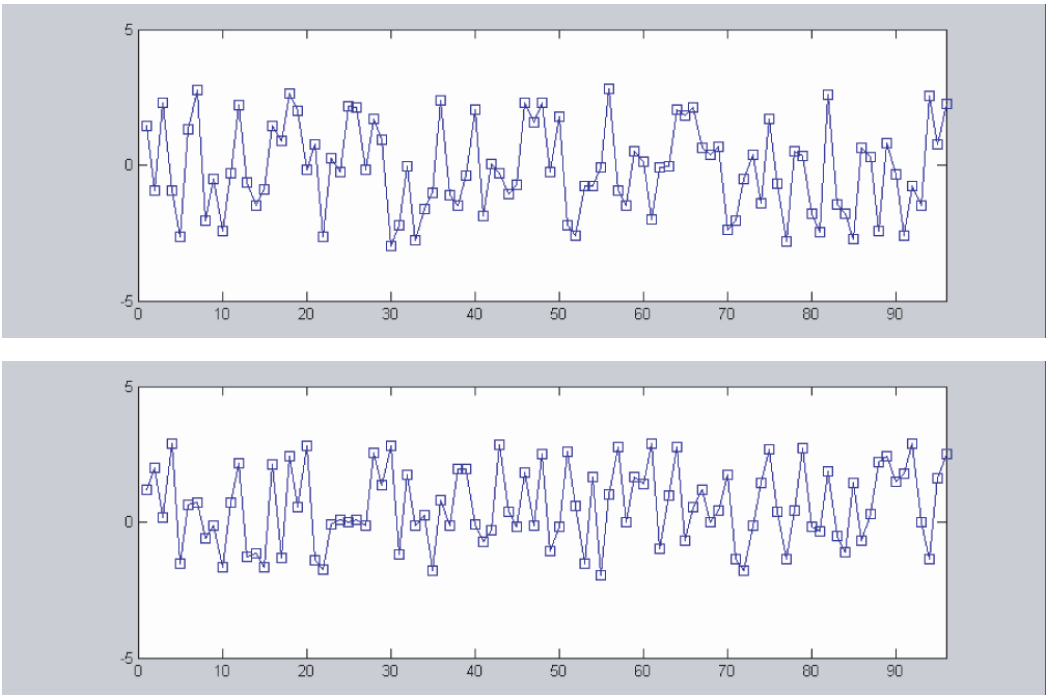


Fig. 12. Accuracy of our interaction moment detection scheme with two different camera configurations.

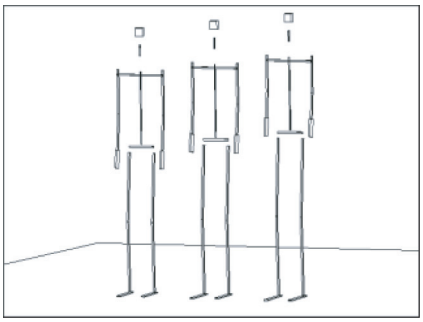


Fig. 13. Three articulated models with different body segment proportions.

In the second experiment, we analyzed our method using ground-truth motion capture data. We obtained motion capture data for kicking and broad jumping, together with their corresponding TV-quality video streams, and then compared the synthesized results with the captured data. The motion library does not include the ground-truth data. Figures 14 and 15 show the comparison results for kicking and broad jumping, respectively. Due to the low-contrast illumination and monotone black suit designed for motion capture, the patch-based segment tracking method of Ju et al. [1996] could not be used for tracking the ankle, knee, and elbow positions. Instead, we used standard 2D marker tracking scheme of Veenman et al. [1998]. Figures 16 and 17 exhibit the time-varying behaviors of angular differences for selected joints in two motions: kicking and broad jumping: The red, green, and blue color curves indicate the reference, captured (ground-truth), and synthesized motions, respectively.

Table IV. Estimated Data for Joint Orientations and Segment Proportions

		Joint Orientation Difference (degree)			Segment Proportion		
		Minimum	Maximum	Average	Actual	Estimated	% error
Middle model	upper arm	0.21	0.91	0.68	2.08	2.06	0.96
	lower arm	0.41	1.10	0.86	1.53	1.51	1.31
	upper leg	0.35	2.21	0.85	3.60	3.57	0.83
	lower leg	0.43	1.27	0.97	3.64	3.58	1.65
Left model	upper arm	0.19	1.32	0.85	1.77	1.79	1.12
	lower arm	0.35	1.54	1.27	1.34	1.34	2.29
	upper leg	0.62	1.97	1.50	3.82	3.90	2.09
	lower leg	0.17	1.45	1.13	3.86	3.92	1.55
Right model	upper arm	0.19	1.56	1.28	2.21	2.18	1.36
	lower arm	0.24	1.33	1.00	1.75	1.71	2.29
	upper leg	0.52	1.29	0.63	3.38	3.48	2.96
	lower leg	0.31	2.10	0.79	3.40	3.49	2.65

Joint orientation differences are measured in angular differences between actual and estimated orientations.

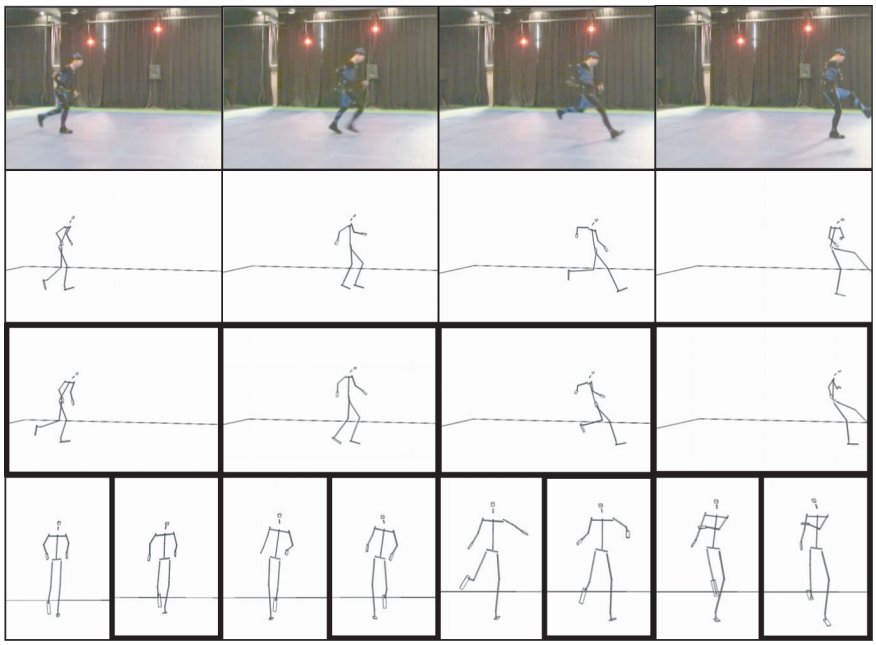


Fig. 14. Captured vs. synthesized motion (kick): the input video (top); captured motion (upper-middle); synthesized motion (lower-middle); poses at a different viewpoints (bottom).

The maximum angular differences between the synthesized motion and ground-truth were less than eight degrees in both motions. For the other motions, the differences were within ten degrees (see Table V). The angular differences in this experiment are relatively larger than those of the previous experiment (performed with the synthetic images) because the marker positions are not coincident with joint positions, unlike the experiment for synthesized data.

In the third experiment, we synthesized various motions of soccer players from videos used for television broadcasting. Figure 18 shows some results on the synthesis of running motions of two different

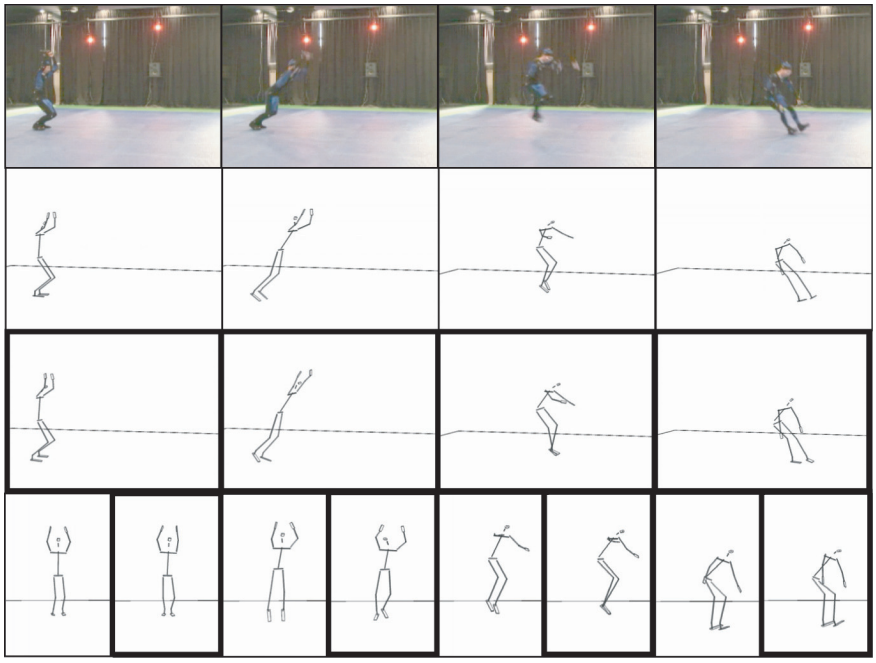


Fig. 15. Captured motion vs. synthesized motion (broad jump with pelvis turning): the input video (top); captured motion (upper-middle); synthesized motion (lower-middle); poses at a different viewpoint (bottom).

soccer players. We used a single reference motion clip to create their running motions, guided by a single input video of 25 frames. Figure 19 shows other results for two different kick motions of soccer players with 51 and 64 frames, respectively. Our scheme successfully synthesized different motions (in accordance with their respective input videos) using a single reference motion clip. Figure 20 shows a header motion of a soccer player that was obtained from a 37-frame video clip. Since the player performed a free flight with the external force of gravity exerted, the root trajectory was adjusted on the basis of the method described in Section 7.2 to keep dynamic balance during the jump.

For the kick motion shown in Figure 19, we performed another experiment to observe how synthesized motions change for different reference motions of the same type. To synthesize the target motion in the video, we use as reference motions four "inside kick" motions with different styles. Let $\{\mathbf{r}_i, 1 \le i \le 4\}$ be the set of reference motions. We tested our scheme with these reference motions to synthesize their corresponding 3D motions $\mathbf{m}_i, 1 \le i \le 4$. We measured the differences between all pairs of motions, including both reference and synthesized motions, as summarized in Table VI. The relative difference $e(\mathbf{m}_a, \mathbf{m}_b)$ between two motions, $\mathbf{m}_a$ and $\mathbf{m}_b$ is given as follows:

$$e(\mathbf{m}_a, \mathbf{m}_b) = \frac{\sum_k \sum_i \left\| \log\left(\mathbf{q}_{a,i}^{-1}(k)\mathbf{q}_{b,i}(k)\right)\right\|}{\max\left\{\sum_k \sum_i \left\| \log\left(\mathbf{q}_{a,i}^{-1}(k)\bar{\mathbf{q}}_i\right)\right\|, \sum_k \sum_i \left\| \log\left(\mathbf{q}_{b,i}^{-1}(k)\bar{\mathbf{q}}_i\right)\right\| t\right\}} \times 100, \tag{25}$$

where $\mathbf{q}_{a,i}$ and $\mathbf{q}_{b,i}$ denote the orientations of joint i at frame k in motions $\mathbf{m}_a$ and $\mathbf{m}_b$, respectively. Here, $\bar{\mathbf{q}}_i$ denotes the orientation of joint i for a neutral posture of the character. The standing posture was chosen as the neutral posture. Even with different reference motions, we observed that the differences among their synthesized motions were quite small (within 3 percent), compared to those among their corresponding reference motions (larger than 12 percent). This suggests that our scheme is not heavily

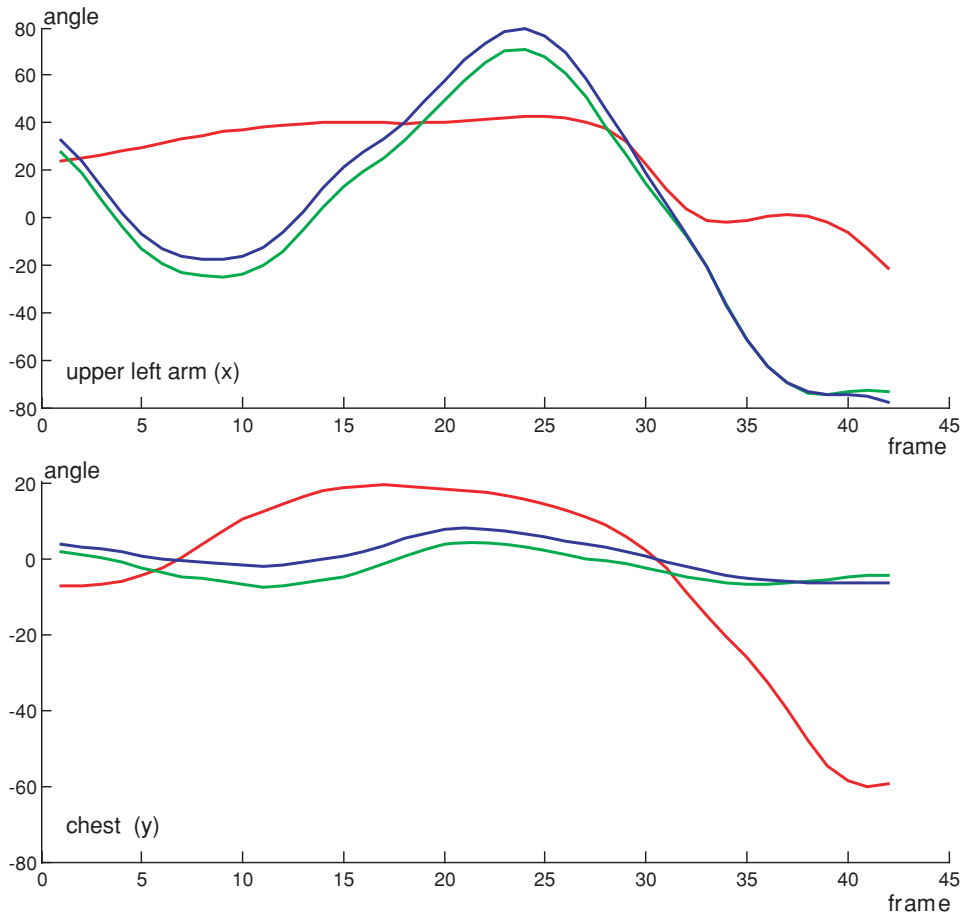


Fig. 16. Joint angle comparison between captured and synthesized motion (kick): upper left arm angle about the local x axis (top) and chest angle about the local y axis (bottom). The red, green, and blue curves indicate the reference, captured (ground-truth), and synthesized angles, respectively.

dependent on the choice of the reference motions, as long as they are of the same type. Furthermore, it implies that we need not prepare an excessive number of reference motions in advance.

To observe how synthesized motion depends on input features, we applied our scheme to the input video using different subsets of the marked features. These subsets were obtained by randomly sampling the features according to a uniform distribution. For each subset, we synthesized its corresponding target motion as specified in the video. We measured the similarity between this motion and the motion synthesized using all input features. The similarity $s(\mathbf{m}_a, \mathbf{m}_a)$ between two different motions $\mathbf{m}_a$ and $\mathbf{m}_b$ is measured by $s(\mathbf{m}_a, \mathbf{m}_a) = 100 - e(\mathbf{m}_a, \mathbf{m}_a)$, where $e(\cdot)$ is defined in Eq. (25). Figure 21 shows the relationship between the similarity and percentage of the sampled features used for motion synthesis. When we used more than 65 percent of the input features, the resulting motions were almost identical to motion synthesized using all features, which supports our spacetime formulation only with unoccluded input features. As the number of available features increased, the synthesized motion more closely resembled the input video. On the other hand, the synthesized motion more closely resembled the reference motion as the number of available features decreased.

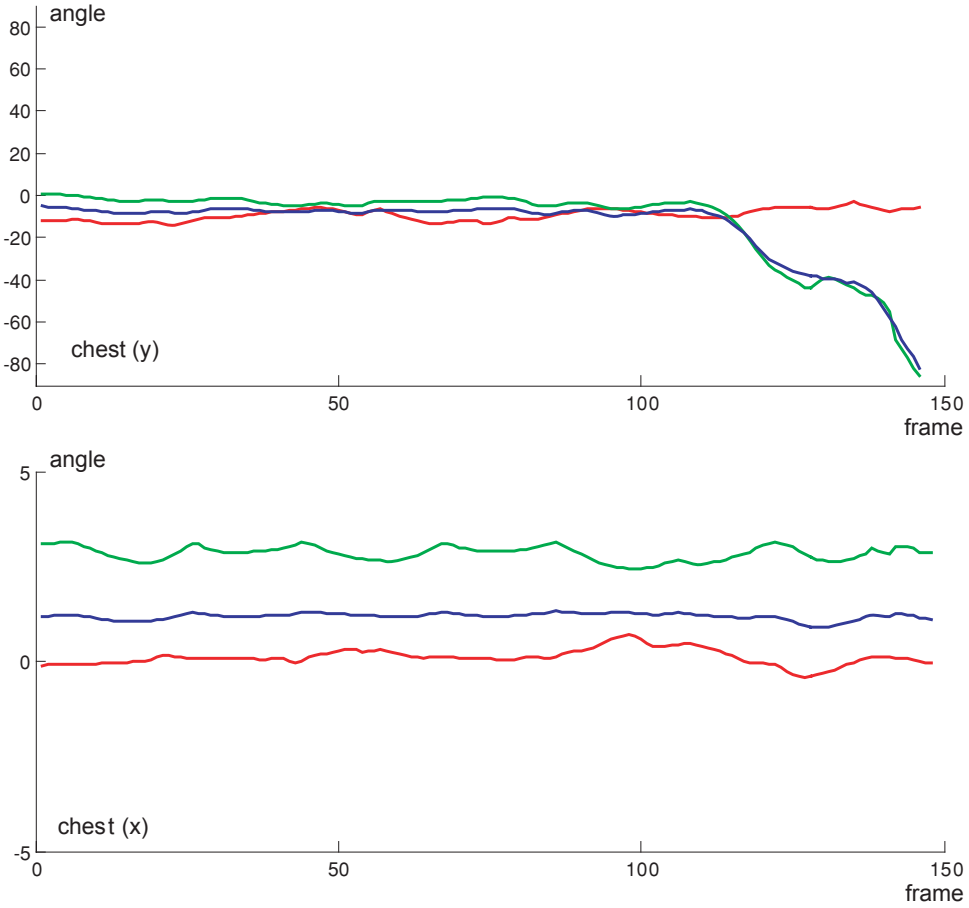
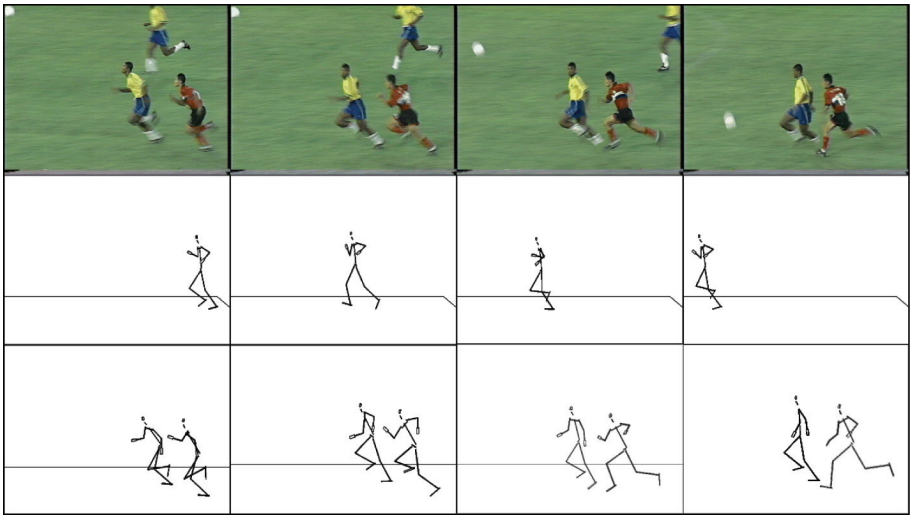


Fig. 17. Joint angle comparison between captured and synthesized motion (broad jump with pelvis turning): chest angle about the local y axis (top) and chest angle about the local x axis (bottom). The red, green, and blue curves indicate the reference, captured (ground-truth), and synthesized angles, respectively.

Table V. Joint Orientation Differences Between Synthesized and Captured (Ground Truth) Motion

	Maximum (Degree)	Average (Degree)
walk	5.54	2.11
run	6.32	4.25
inside kick	8.26	4.78
instep kick	9.21	4.33
broad jump	7.30	3.38
broad jump with pelvis turning	8.47	4.97

In the final experiment, we show the potential of our scheme in motion capture. To explore this possibility, a precaptured broad jump was chosen as the reference motion, in which the performer swings his arms back and forth twice while jumping. As shown in Figure 22, we took image sequences of two different broad jumps performed by a nonprofessional subject, using a video camera in a room without any special lighting. These motions were highly dynamic and quite different from each other: a normal



(a) the input video (top); timewarped reference motion (middle); and results (bottom)

(b) a posture sequence for the red player

(c) a posture sequence for the yellow player

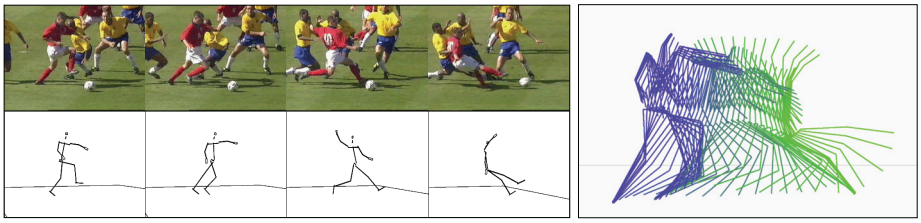
Fig. 18. Running motions.

broad jump similar to the reference motion and one with pelvis turning. Each of the corresponding image sequences consists of 162 frames sampled at 29.97 frames per second. The motions reconstructed from these image sequences are visually convincing. In fact, our scheme has demonstrated its capability to capture a rather broad range of variants with a given reference motion.

Table VII shows timing data for our optimization method. Here, t_1 denotes the average time per iteration of computing camera parameters and segment proportions, and t_2 is that of computing joint orientations. The total time includes the sum of computation time by alternating steps for all iterations, together with some overhead for combining these two steps. The maximum number of alternating iterations was six, and the total amount of CPU time was less than one minute. In all experiments, the whole process of each motion synthesis was done in less than five minutes, including interactive feature marking. Even with a video of low sampling rate (29.97 Hz), our scheme produced highly dynamic motion by exploiting a densely, sampled reference motion clip (60 Hz).

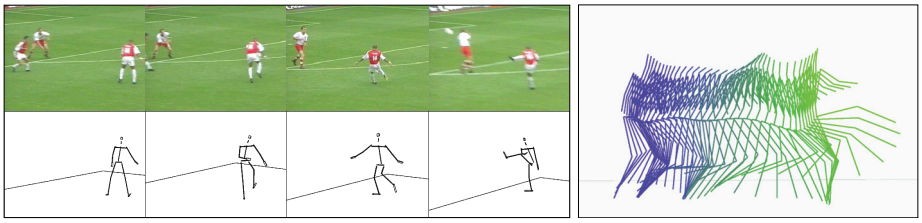
9. DISCUSSION

In this section, we discuss the weaknesses and limitations of our approach, together with the hidden assumptions. These are classified into five categories: motion domain, manual operations, reference motion selection, parameter optimization, and root trajectory estimation.



(a) the input video (top) and synthesized result (bottom)

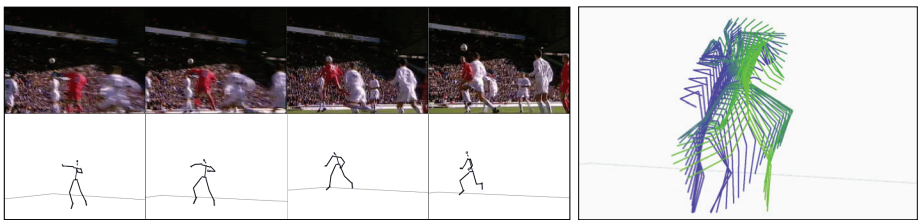
(b) a posture sequence



(c) the input video (top) and synthesized result (bottom)

(d) a posture sequence

Fig. 19. Kick motions.



(a) the input video (top) and synthesized result (bottom)

(b) a posture sequence

Fig. 20. Header motion.

Motion Domain. We assume that the camera always points at the root joint to keep it at the center of the image plane. Therefore, the 2D joint positions of the input video are in an ideal situation for feature tracking when the sagital plane of the character is parallel with the image plane at every frame of the input video. The class of motions satisfying this situation covers locomotive motions and its variations, such as jumping and kicking, wherein not much rotation about the global-up axis of the character is involved. If the sagital plane is perpendicular to the image plane, our system does not work, even for these motions, since their 2D projections yield rather small joint position variations over time. Fortunately, the two planes are rarely perpendicular in sports videos, except for close-up views.

Manual Operations. The main sources of manual operations are feature tracking and keytime extraction. In our experiments, 20∼30 percent of unoccluded features are manually marked due to weak frame coherency, abrupt illumination changes, and cluttered backgrounds. This ratio will generally be

Table VI. Motion Differences Between Synthesized and Reference Motions

	$\mathbf{r}_1$	$\mathbf{r}_2$	$\mathbf{r}_3$	$\mathbf{r}_4$	$\mathbf{m}_1$	$\mathbf{m}_2$	$\mathbf{m}_3$	$\mathbf{m}_4$
$\mathbf{r}_1$	0	12.1	18.4	23.1	24.1	23.8	24.0	23.5
$\mathbf{r}_2$	—	0	19.7	22.0	29.1	28.9	28.3	27.2
$\mathbf{r}_3$	—	—	0	17.0	25.4	25.3	25.9	26.1
$\mathbf{r}_4$	—	—	—	0	18.1	18.8	17.9	19.4
$\mathbf{m}_1$	—	—	—	—	0	1.3	1.9	1.8
$\mathbf{m}_2$	—	—	—	—	—	0	2.3	2.1
$\mathbf{m}_3$	—	—	—	—	—	—	0	1.6
$\mathbf{m}_4$	—	—	—	—	—	—	—	0

$\{\mathbf{r}_i, 1 \leq i \leq 4\}$ and $\{\mathbf{m}_i, 1 \leq i \leq 4\}$ are a set of reference motions and that of the corresponding synthesized motions, respectively.

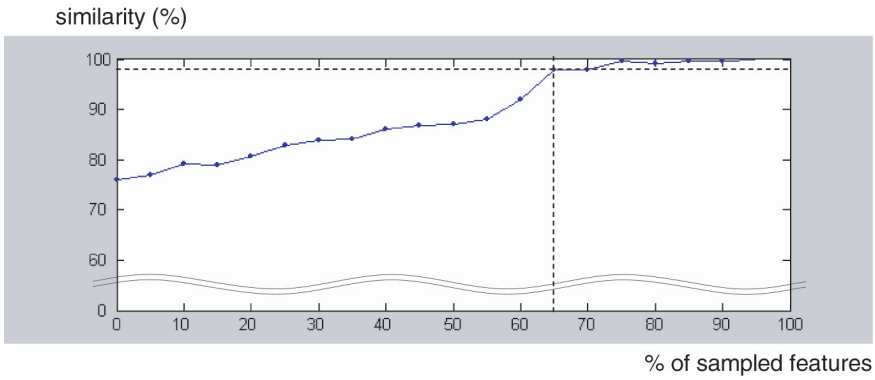
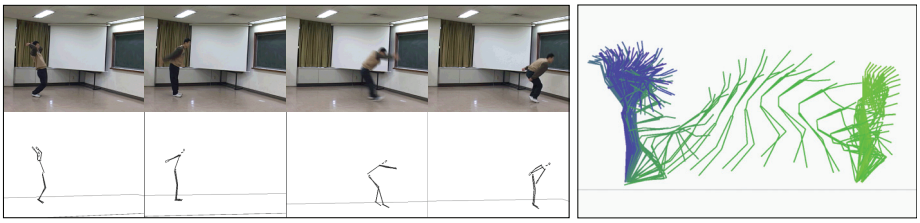


Fig. 21. Synthesized motions with the different subsets of input features. The vertical and horizontal axes indicate the similarity and percentage of the sampled features, respectively.

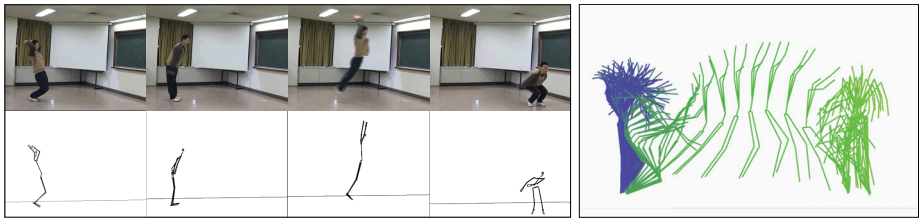
much higher if we include the tracking failure by occlusion of features. As illustrated in Figure 21, our system is not overly sensitive to tracking failure as long as more than 65 percent of the feature points are obtained either automatically or manually because our spacetime formulation does not require all feature points. In fact, the reference motion fills the holes caused by missing features during parameter computation. For keytime extraction, our system works well for those motions that involve interactions between the ground plane and feet, in both 2D video streams and 3D motion clips. However, other keytimes, such as instances of touching a ball with a foot and the head, are specified manually.

Reference Motion Selection. The quality of a synthesized motion is not very sensitive to the reference motion, as long as they share the same motion type, as demonstrated in Table IV. Exploiting this fact, the repertoire of motions in a motion library could be optimized, as well as the discretization of the camera configuration. Finally, as the repertoire of the motion library is rich, the current exhaustive search for a reference motion by votes will not scale well with growth of the library. A more systematic way to access the reference motion should be developed. We leave these issues for future research.

Parameter Optimization. To compute the local posture at each frame, we find optimal parameters by alternately optimizing the objective function given in Eq. (13); once to optimize local parameters, such as joint orientations (while fixing the global parameters, such as camera parameters and segment proportions), and once in the symmetrical way. Our experience shows that this strategy works extremely well, which resolves the slow convergency problem caused by the simultaneous optimization for all parameters. The fast convergency, we believe, is due to the reference motion and segment proportion distribution function, which guides their respective optimization steps.



(a) broad jump (b) a posture sequence



(c) broad jump with pelvis turning (d) a posture sequence

Fig. 22. Broad jumps.

Table VII. Timing Data (CPU Time in Seconds)

	# of Frames	# of Iterations	t_1	t_2	Total Time
Running (red player)	25	4	0.84	1.32	8.67
Running (yellow player)	25	4	0.72	1.06	7.23
Kicking #1	52	6	0.65	2.43	18.60
Kicking #2	48	6	0.71	2.92	21.79
Heading	37	5	0.84	1.43	11.36
Broad jumping	162	5	0.78	5.11	33.46
Broad jumping with pelvis turning	162	5	0.89	6.05	36.97

Root Trajectory Estimation. For motions involving interactions between the ground plane and feet, our system works well, unless the synthesized motion is overly timewarped. For motions with a flight phase, such as jumping, we have to manually provide a scale factor η to estimate the root trajectory using Eq. (23). At the moment, we find η by trial and error. In particular, it is difficult to find η when a subject jumps toward the camera. A better way would be to manually specify the peak point of the flight phase to automatically derive η, which we leave as another topic for future research.

10. CONCLUSION

In this article, we have presented a practical, semiautomatic method for synthesizing human motions from a single video stream by using a motion library. Choosing a 3D motion in the library as a reference, we resolve the inherent depth ambiguity in motion synthesis. The synthesized motion is smooth, even for a video stream with noisy features. Moreover, our approach can handle highly dynamic motions with weak frame coherence in the input video stream. Provided with a feasible set of motions as a library, our method can be used to construct a wide variety of motions in real situations, such as sports events and dance performances.

Within the entire process of our motion synthesis, we try to minimize user interactions. In feature tracking, we take a semiautomatic scheme to track features by adopting a previous patch-based human body tracking method. The reference motion is chosen from these input features while an estimate of appropriate camera parameters is made simultaneously. For locomotive motions, we automatically find the interaction moments in the input video, as well as in the reference motion, for our keytime-based timewarping scheme.

We have demonstrated the effectiveness of our method by capturing various motions from real videos. Our motion could be used as a simple, effective alternative to motion capture. That is, our scheme can be employed to derive the desired motion from a monocular video captured by a hand-held camera, even outside a motion capture studio.

REFERENCES

BARRON, C. AND KAKADIARIS, I. A. 2000. Estimating anthropometry and pose from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 669–676.

BEN-ARIE, J., PNADIT, P., AND RAJARAM, S. 2001a. Design of a digital library for human movement. In *Proceedings of the 1st ACM/IEEE-CS Joint Conference on Digital Libraries*. 300–309.

BEN-ARIE, J., PNADIT, P., AND RAJARAM, S. 2001b. View-Based human activity recognition by indexing and sequencing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 78–83.

BRAND, M. 1999. Shadow puppetry. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

BREGLER, C., LOEB, L., CHUANG, E., AND DESHPANDE, H. 2002. Turning to the masters: Motion capturing cartoons. *Comput. Graph. 36*, 399–407.

BREGLER, C. AND MALIK, J. 1998. Tracking people with twists and exponential maps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8–15.

BREGLER, C., MALIK, J., AND PULLEN, K. 2004. Twist-Based acquisition and tracking of animal and human kinematics. *Int. J. Comput. Vision 56*, 3, 179–194.

BRUDERIN, A. AND WILLIAMS, L. 1995. Motion signal processing. *Comput. Graph. 29*, 4, 97–104.

COHEN, M. F. 1992. Interactive spacetime control for animation. *Comput. Graph. 26*, 2, 293–302.

CORMEN, T. H., LEISERSON, C. E., AND RIVEST, R. L. 1999. *Introduction to Algorithms*. MIT Press, Cambridge, MA.

DELAMARRE, Q. AND FAUGERAS, O. 2001. 3d articulated models and multiview tracking with physical forces. *Comput. Vision Image Understanding 81*, 3, 328–357.

DEMIRDJIAN, D., KO, T., AND DARRELL, T. 2003. Constraining human body tracking. In *Proceedings of the International Conference on Computer Vision*.

DEMORI, R. AND PROBST, D. 1986. *Handbook of Pattern Recognition and Image Processing*, 1st ed. Academic Press, London.

DIFRANCO, D. E., CHAM, T. J., AND REHG, J. M. 1999. Recovery of 3D articulated motion from 2D correspondences. Tech. Rep. TR-CRL-99-7, Cambridge Research Laboratory.

DRUMMOND, T. AND CIPOLLA, R. 2001. Real-Time tracking of highly articulated structures in the presence of noisy measurements. In *Proceedings of the International Conference on Computer Vision*. 315–320.

FANG, A. C. AND POLLARD, N. C. 2003. Efficient synthesis of physically valid human motion. *Comput. Graph. 37*.

FILMBOX. *Metamotion*. www.metamotion.com.

FLETCHER, R. 1980. *Practical Methods of Optimization*, 1st ed. Wiley and Sons, New York.

GAVRILA, D. M. 1999. The visual analysis of human movement: A survey. *Comput. Vision. Image Understanding 3*, 1, 82–98.

GILL, P. E. AND MURRAY, W. 1974. *Numerical Methods for Constrained Optimization*, 1st ed. Academic Press, London.

GLEICHER, M. 1997. Motion editing with spacetime constraints. In *Proceedings of the Symposium on Interactive 3D Graphics*. 139–148.

GLEICHER, M. 1998. Retargetting motion to new characters. *Comput. Graph. 32*, 33–42.

GLEICHER, M. AND FERRIER, N. 2002. Evaluating video-based motion capture. In *Proceedings of the Computer Animation Conference*. 75–80.

GROCHOW, K., MARTIN, S. L., HERTZMANN, A., AND POPOVIC, Z. 2004. Style-Based inverse kinematics. *Comput. Graph. 38*, 522–531.

HOWE, N. R., LEVENTON, M. E., AND FREEMAN, W. T. 2000. Bayesian reconstruction of 3D human motion from single-camera video. *Advances in Neural Information Processing Systems*. 820–826.

JU, S. X., BLACK, M. J., AND YACOOB, Y. 1996. Cardboard people: A parameterized model of articulated image motion. In *Proceedings of the 2nd International Conference on Automatic Face and Gesture Recognition*.

KAKADIARIS, I. AND METAXAS, D. 1996. Model-Based estimation of 3D human motion with occlusion based on active multi-viewpoint selection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 81–89.

KIRK, A. G., O'BRIEN, J. F., AND FORSYTH, D. A. 2005. Skeletal parameter estimation from optical motion capture data. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 782–788.

LEE, J., CHAI, J., REITSMA, P., HODGINS, J., AND POLLARD, N. 2002. Interactive control of avatars aniamted with human motion data. *Comput. Graph. 36*, 491–500.

LEE, J. AND SHIN, S. Y. 1999. A hierarchical approach to interactive motion editing for human-like figures. *Comput. Graph. 33*, 39–48.

LEE, S., WOLBERG, G., AND SHIN, S. Y. 1997. Scattered data interpolation with multilevel b-splines. *IEEE Trans. Visualization Comput. Graph. 3*, 3, 228–244.

LEUNG, M. K. AND YANG, Y. H. 1995. First sight: A human body outline labeling system. *IEEE Trans. Pattern Anal. Mach. Intell. 17*, 4, 359–377.

LIEBOWITZ, D. AND CARLSSON, S. 2001. Uncalibrated motion capture exploiting articulated structure constraints. In *Proceedings of the 8th International Conference on Computer Vision*. 230–237.

LIU, C. K. AND POPOVIĆ, Z. 2002. Synthesis of complex dynamic character motion from simple animations. *Comput. Graph. 36*, 4, 408–416.

MOESLUND, T. B. AND GRANUM, E. 2001. A survey of computer vision-based human motion capture. *Comput. Vision Image Understanding 81*, 3, 231–268.

MORRIS, D. D. AND REHG, J. 1998. Singularity analysis for articulated object tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

NOMA, T., OISHI, K., FUTSUHARA, H., BABA, H., OHASHI, T., AND EJIMA, T. 1999. Motion generator approach to translating human motion from video to animation. In *Proceedings of the Pacific Graphics Conference*. 50–59.

PARK, M. J., CHOI, M. G., AND SHIN, S. Y. 2002. Human motion reconstruction from inter-frame feature correspondences of a single video stream. In *Proceedings of the ACM Symposium on Computer Animation*. 113–120.

PAVLOVIC, V., REHG, J. M., CHAM, T. J., AND MURPHY, K. 1999. A dynamic Bayesian network approach to figure tracking using learned dynamic models. In *Proceedings of the International Conference on Computer Vision 1*, 94–101.

PHEASANT, S. 1996. *Bodyspace: Anthropometry, Ergonomics and the Design of Work*, 2nd ed. Taylor and Francis, Bristol, PA.

PLAENCKER, R. AND FUA, P. 2001. Tracking and modeling people in video sequences. *Comput. Vision Image Understanding 81*, 3, 285–302.

POPOVIĆ, Z. AND WITKIN, A. 1999. Physically-Based motion transformation. *Comput. Graph. 33*, 11–20.

POSER. *Curious Labs*. www.curiouslabs.com.

PRESS, W. H., TEUKOLSKY, S. A., VETTERLING, W. T., AND FLANNERY, B. P. 1992. *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed. Cambridge University Press, New York.

RAMANAN, D. AND FORSYTH, D. A. 2003. Automatic annotation of everyday movements. *Neural Information Processing Systems (NIPS)*.

REHG, J. M. AND KANADE, T. 1994. Visual tracking of high DOF articulated structures: An application to human hand tracking. In *Proceedings of the European Conference on Computer Vision*. 35–46.

ROSE, C., GUENTER, B., BODENHEIMER, B., AND COHEN, M. F. 1996. Efficient generation of motion transitions using spacetime constraints. *Comput. Graph. 30*, 147–154.

SAFONOVA, A., HODGINS, J., AND POLLARD, N. 2004. Synthesizing physically realisitic human motion in low-dimensional, behavior-specific spaces. *Comput. Graph. 38*.

SHOEMAKE, K. 1985. Animating rotation with quaternion curves. *Comput. Graph. 19*, 245–254.

SIDENBLADH, H., BLACK, M. J., AND FLEET, D. J. 2000a. Stochastic tracking of 3D human figures using 2D image motion. In *Proceedings of the European Conference on Computer Vision*. 702–718.

SIDENBLADH, H., BLACK, M. J., AND FLEET, D. J. 2000b. Stochastic tracking of 3d human figures using 2d image motion. In *Proceedings of the European Conference on Computer Vision*.

SIDENBLADH, H., BLACK, M. J., AND SIGAL, L. 2002. Implicit probabilistic models of human motion for synthesis and tracking. In *Proceedings of the European Conference on Computer Vision*. 784–800.

SMINCHISESCU, C. AND TRIGGS, B. 2001. Covariance scaled sampling for monocular 3D body trakcing. In *Proceedings of the IEEE Conference Computer Vision and Pattern Recognition*. 447–454.

SONG, Y., GONCALVES, L., AND PERONA, P. 2003. Unsupervised learning of human motion. *IEEE Trans. Pattern Anal. Mach. Intell. 25*, 7, 814–827.

STARCK, J. AND HILTON, A. 2003. Model-Based multiple view reconstruction of people. In *Proceedings of the International Conference on Computer Vision*.

TAK, S., SONG, O., AND KO, H. 2000. Motion balancing filtering. *Comput. Graph. Forum 9*, 3, 437–446.

TAYLOR, C. J. 2000. Reconstruction of articulated objects from point correspondences in a single uncalibrated image. *Comput. Vision Image Understanding 80*, 8, 349–363.

TIAN, T.-P., LI, R., AND SCLAROFF, S. 2005. Articulated pose estimation in a learned smooth space of feasible solutions. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2005*.

URTASUN, R., FLEET, D., AND FUA, P. 2005. Monocular 3-d tracking of the golf swing. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2005*.

VEENMAN, C. J., HENDRIKS, E. A., AND REINDERS, M. J. T. 1998. A fast and robust point tracking algorithm. In *Proceedings of the International Conference on Image Processing*.

WANG, L., HU, W., AND TAN, T. 2003. Recent development in human motion analysis. *Pattern Recogn. 36*, 585–601.

WITKIN, A. AND KASS, M. 1988. Spacetime constraints. *Comput. Graph. 22*, 4, 159–168.

WITKIN, A. AND POPOVIĆ, Z. 1995. Motion warping. *Comput. Graph. 29*, 105–108.

WREN, C., AZARBAYEJANI, A., AND PENTLAND, A. 1997. Pfinder: Real-Time 3-D tracking of the human body. *IEEE Trans. Pattern Anal. Mach. Intell. 19*, 7, 780–785.

YAMANE, K., KUFFNER, J., AND HODGINS, J. 2004. Synthesizing animations of human manipulation tasks. *Comput. Graph. 38*.

ZHENG, J. Y. AND SUEZAKI, S. 1998. A model-Based approach in extracting and generating human motion. In *Proceedings of the 14th International Conference on Pattern Recognition*. 1201–1205.

Received January 2005; revised July 2006; accepted August 2006