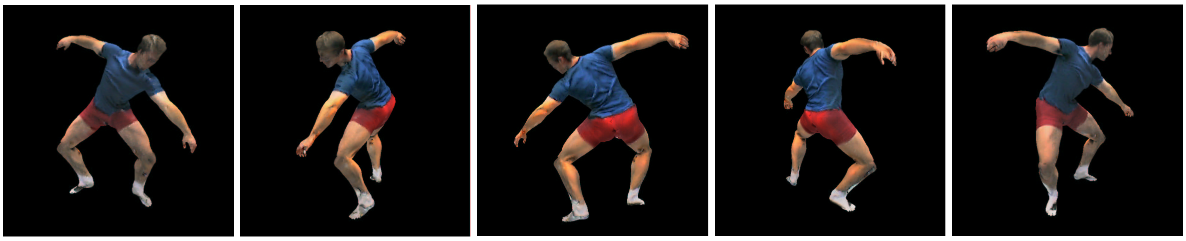
ACM Transactions on Graphics, vol. 22, no. 3, pp. 569–577, July 2003 (Proc. ACM Siggraph 2003, San Diego, USA)

Free-Viewpoint Video of Human Actors

Joel Carranza Christian Theobalt Marcus A. Magnor Hans-Peter Seidel
MPI Informatik
Saarbrücken, Germany
{*carranza, theobalt, magnor, hpseidel*}*@mpi-sb.mpg.de*



Abstract

In free-viewpoint video, the viewer can interactively choose his viewpoint in 3-D space to observe the action of a dynamic real-world scene from arbitrary perspectives. The human body and its motion plays a central role in most visual media and its structure can be exploited for robust motion estimation and efficient visualization. This paper describes a system that uses multi-view synchronized video footage of an actor's performance to estimate motion parameters and to interactively re-render the actor's appearance from any viewpoint.

The actor's silhouettes are extracted from synchronized video frames via background segmentation and then used to determine a sequence of poses for a 3D human body model. By employing multi-view texturing during rendering, time-dependent changes in the body surface are reproduced in high detail. The motion capture subsystem runs offline, is non-intrusive, yields robust motion parameter estimates, and can cope with a broad range of motion. The rendering subsystem runs at real-time frame rates using ubiquous graphics hardware, yielding a highly naturalistic impression of the actor. The actor can be placed in virtual environments to create composite dynamic scenes. Free-viewpoint video allows the creation of camera fly-throughs or viewing the action interactively from arbitrary perspectives.

CR Categories: I.4.8 [Image Processing and Computer Vision]: Scene Analysis—Motion,Shape,Time-Varying Imagery,Tracking; I.4.9 [Image Processing and Computer Vision]: Applications;I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation;

Keywords: human motion capture, body model, multi-video texturing, image-based rendering

1 Introduction

Currently, visual media such as television and motion pictures only present a two dimensional impression of the real world. The camera positions are unchangeable and determined only by the director. Traditionally, the goal of Computer Graphics research has been to develop algorithms for realistic rendering of synthetic scenes from arbitrary viewpoints. The focus of Computer Vision, on the other hand, is the inverse process of extracting a model of a given real-world scene using information from optical sensors. In recent years, the advent of new technologies and challenges from possible new applications have led to a convergence of both fields [Terzopoulos et al. 1995]. One interdisciplinary research area that embraces developments from both disciplines is *Free-Viewpoint Video* [Wuermlin et al. 2002; Matsuyama and Takai 2002]. The goal is to bring about a sense of immersion by giving the viewer the freedom to choose his viewpoint at will, displaying a dynamic real-world scene from arbitrary perspectives.

The possible applications are manifold. A free-viewpoint video system can assist a coach in analyzing the efficiency of his athlete's motion. Commentators in a post-game analysis of sports events are provided with a powerful tool to show from moving viewpoints a basketball player jumping towards the hoop.

Motion pictures gain an interactive and immersive dimension by seating the viewer in the director's chair and letting him choose his viewpoint interactively. For conventional movie production, free-viewpoint techniques offer new tools for the post-production process, including camera movement and virtual reality elements.

In most movie scenes, attention focuses on the actors involved. In recent feature films, free-viewpoint video elements involving actors, such as *freeze-and-rotate* camera shots have been included. These effects are made possible by recording the actor with tens to hundreds of cameras placed around the set. Unfortunately, this hardware effort is only affordable for high budget movie productions and the large number of video streams are not a feasible data format for distribution of free-viewpoint video as mass media.

Due to the importance of actors for visual media, there is a strong motivation to make free-viewpoint video acquisition of human scenes less cumbersome. The produced data format must be suitable for effective transmission and real-time rendering on state-of-the-art consumer-grade hardware.

Previous approaches for free-viewpoint rendering of real-world scenes presented in the Computer Graphics community typically involve the explicit reconstruction of scene geometry from the images at every time instant [Matusik et al. 2001; Matusik et al. 2000;

Moezzi et al. 1997] without using a priori model information and explicit representation of motion data. Researchers in Computer Vision have developed marker-free motion capture algorithms that employ a priori body models for tracking [Gavrila 1999]. Using the data to generate realistic novel views of a scene, however, is usually not addressed.

We believe that the combination of marker-free motion capture and multi-view texture generation is highly effective for synthesizing novel views of a human in motion. This paper presents a new approach that implements this design concept in a working system. A generic body model consisting of a triangle mesh shape representation and a kinematic skeleton is used to follow the motion over time. The input to the system consists of synchronized multi-view video streams that are recorded in a controlled environment by stationary video cameras. Silhouette images of the person are extracted in each camera view through background subtraction. The silhouettes form the input to the presented motion capture algorithm. The motion parameters, i.e rigid body transformations between adjacent body segments, are found at each time step in an offline procedure by optimizing the overlap between the projected model silhouettes and the input image silhouettes. Using this method, motion tracking becomes possible without any intrusion into the recorded environment. The motion parameter estimation is highly robust and can handle a broad range of motion. As will be demonstrated, this enables the system to correctly recover even such complex motion as ballet dance.

The same model is used for both motion capture and rendering. During replay of the recorded sequence from an arbitrary viewpoint, the image data from all input cameras is used to generate realistic time-dependent surface textures. Both the rendering, as well as the numerical computations involved in motion capture, make effective use of programmable features on today's graphics boards.

The rest of this paper proceeds with a review of related work and a comparison of our method to relevant approaches in Sect. 2. In Sect. 3 the acquisition environment used to record the multi-view video sequences is described. The body model employed is explained thereafter in Sect. 4. The motion capture subsystem is explained in detail in Sect. 5. Free-viewpoint rendering and texture generation are presented in Sect. 6, and Sect. 7 presents results obtained using the system. The paper concludes in Sect. 8 with a discussion of features and limitations of the presented system and gives an outlook to future work.

2 Related Work

The review of previous work begins with a brief summary of research in human motion capture presented in the Computer Vision literature. Thereafter, relevant work on scene reconstruction and free-viewpoint rendering in image-based Computer Graphics is discussed.

2.1 Human Motion Capture

Human Motion Capture is the process of acquiring the parameters of human motion. Commercial human motion capture systems can be classified as mechanical, electromagnetic, or optical systems [Menache 1995]. Video-based systems used in the industry typically require the person to wear optical markers on the body. The 3D marker locations are used to fit a kinematic skeleton to the motion data [Silaghi et al. 1998]. During the acquisition of video sequences for free-viewpoint video, no intrusion into the scene can be tolerated.

In Computer Vision, algorithms for marker-free optical motion capture have been developed [Gavrila 1999]. Some methods work only in 2D and represent the body by a probabilistic region model [Wren et al. 1997] or a stick figure [Leung and Yang 1995]. More advanced algorithms employ a kinematic body model assembled of simple shape primitives, such as cylinders [Rohr 1993], ellipsoids [Cheung et al. 2000], or superquadrics [Gavrila and Davis 1996]. Inverse kinematics approaches linearize around the non-linear mapping from image to parameter space [Bregler and Malik 1998; Yonemoto et al. 2000] to compute model parameters directly. Analysis-through-synthesis methods search optimal body parameters that minimize the misalignment between image and projected model [Martinez 1995]. To estimate the goodness-of-fit, features such as image discontinuities are typically extracted from the video frames [Gavrila and Davis 1996].

Recently, it has been shown that the real-time reconstruction of object volumes from silhouette images is possible [Borovikov and Davis 2000]. In [Cheung et al. 2000] an expectation-maximization-like algorithm is used to fit an ellipsoidal model to human body volumes in real-time. A force field exerted by the voxels is used in [Luck and Small 2002] to fit a kinematic skeleton to volume data at interactive frame rates. A combination of feature tracking and volume reconstruction can be used to fit a multi-layer skeleton to human motion data [Theobalt et al. 2002]. Other approaches run off-line and fit a pre-defined kinematic model with triangular mesh surface representation [Bottino and Laurentini 2001] to the volumes by minimizing a distance metric or making use of Kalman Filter-based tracking [Mikić et al. 2001]. Unfortunately, the employed body models are too simple to be suitable for rendering. Additional inaccuracies are introduced because the shape reconstructed from image silhouettes is only a coarse approximation to the actual body shape [Laurentini 1994].

Algorithms that compute motion parameters from silhouette images directly prevent computationally expensive scene reconstruction. In [Delamarre and Faugeras 1999], a body model assembled of primitive shapes is aligned with input image silhouettes by means of a force field exerted on the model edges in the image plane.

A sophisticated body model that represents surface deformations using implicit surfaces is fitted to video data in [Plaenkers and Fua 2001] by using depth and silhouette information. Unfortunately, stereo methods have stricter visibility requirements and have not demonstrated that they are capable of handling as broad a range of motion as our method. In [Allen et al. 2002], upper body deformations are modeled by interpolating between real body scans using a displaced subdivision surface. This method produces highly realistic geometric body models but the setup used for acquisition is very complex and would be difficult to incorporate into a free-viewpoint video system.

The method presented in this paper applies the same detailed body model for motion capture as well as for rendering. Tracking is performed by optimizing the overlap between the model silhouette projection and input silhouette images in all camera views. The algorithm is insensitive to inaccuracies in the silhouettes and does not suffer from robustness problems as they commonly occur in many feature-based motion capture algorithms. The fitting procedure works in the image plane only, reconstruction of scene geometry is not required. Many marker-free video-based motion capture methods impose significant constraints on the allowed body pose or the tractable direction of motion. In contrast, our system handles a broad range of body gestures. Even fast motion is robustly recovered. Furthermore, our motion capture algorithm can make effective use of modern graphics processors by delegating the error metric evaluation to the graphics board.

2.2 Image-Based Modeling and Rendering

The approaches described in the previous section deal primarily with robust computation of human motion parameters. The rendering of human motion from arbitrary viewpoints is not their main objective. Quite different in focus is the field of image-based rendering and reconstruction, whose goal is to generate novel views of a scene from real input images [Levoy and Hanrahan 1996].

The reconstruction of scene models from static images is commonly referred to as *3D photography* [Curless and Seitz 2000]. Most algorithms falling into this category can also be applied to dynamic scenes. A prominent class of geometry reconstruction algorithms are shape-from-silhouette approaches. These methods derive the geometry of a foreground object from its silhouette views and can, at best, reconstruct the visual hull, a coarse approximation to the actual scene geometry [Laurentini 1994]. A polygonal reconstruction of a person's shape from multiple silhouettes at interactive frame rates is shown in [Matusik et al. 2001]. The derivation of colored voxel models of dynamic scenes for free-viewpoint rendering is also an option [Moezzi et al. 1997]. Multi-view video, voxel-based reconstruction, and space-time interpolation along the 3D scene flow can be applied to create models of moving human actors at intermediate time steps between consecutive sets of multi-view video frames [Vedula et al. 2002]. In [Matusik et al. 2000] novel views of a person's visual hull are generated in real-time from silhouette views by performing computations purely in the image plane, thereby avoiding explicit 3D reconstruction.

In [Matsuyama and Takai 2002], a polygonal representation of a person's visual hull is computed and view-dependent texturing is used to generate a naturalistic surface appearance in an off-line process. A point-based representation of a person is reconstructed from multiple cameras using image-based visual hulls in [Wuermlin et al. 2002]. The point data is encoded using hierarchical space partitioning. Basic viewing functions for 3D video are also provided. The application of stereo-algorithms to reconstruct the geometry of dynamic scenes has also been considered. In [Narayanan et al. 1998], a dome of 51 cameras was used to reconstruct scene geometry via dense stereo. The computation of geometry models of humans via multi-camera stereo and their transmission over a network has also been a component of tele-presence applications [Mulligan and Daniilidis 2000].

In contrast to the methods described above, our system employs an a priori shape model that is adapted to the observed person's outline. Shape-from-silhouette methods exhibit visually disturbing geometry errors in the form of phantom volumes or quantization artifacts. These geometry artifacts can not occur in our system. Furthermore, because graphics processors are optimized for polygon visualization, a triangle based shape model is better suited for rendering on common graphics hardware than a volumetric model.

Stereo approaches need a comparably high number of recording cameras in order to reconstruct high quality scene models. In addition, because the correspondence problem in the image plane is difficult to solve, these approaches lack robustness. In contrast, our system produces high quality 3D representations with only eight cameras.

Finally, the outputs of our system are particularly suitable for transmission over bandwidth-limited network connections. For every time instant, video frames and a small number of motion parameters would need to be transferred. The model geometry only needs to be transmitted once and 2D video encoding for the textures can easily be applied. The described system fits in the MPEG-4 standard [Koenen 2002] where triangle meshes are defined as media objects and the texture information is encoded using state-of-the-art 2D video codecs. To obtain visual quality similar to that of a triangle mesh, volumetric reconstruction methods must generate highly detailed point datasets for every time instant. Transmission of these large datasets may not be feasible over standard network connections.

3 Multi-View Video Recording

The video sequences used as input to our system are recorded in a multi-view camera studio (Fig. 1). IEEE1394 cameras are placed in a convergent setup around the center of the scene. The video sequences used for this paper are recorded from static viewing positions arranged at approximately equal angles and distances around the center of the room. The cameras are synchronized via an external trigger, and pairs of cameras are controlled by an Athlon 1GHz PC that streams the recorded frames directly to disk. Video frames are recorded at a resolution of 320x240 at 15 fps or at 640x480 at 10 fps. The frame rate is fundamentally limited to 15 fps by the external trigger. At the higher resolution additional I/O-overhead limits the performance. Using Tsai's algorithm [Tsai 1986] the cameras' intrinsic and extrinsic parameters are determined, calibrating every camera into a common global coordinate system. The lighting conditions in the acquisition room are controlled. The influence of external light sources on the set is minimized by black curtains hung from each wall. All cameras are color-calibrated by adapting their white color to a white reference object.

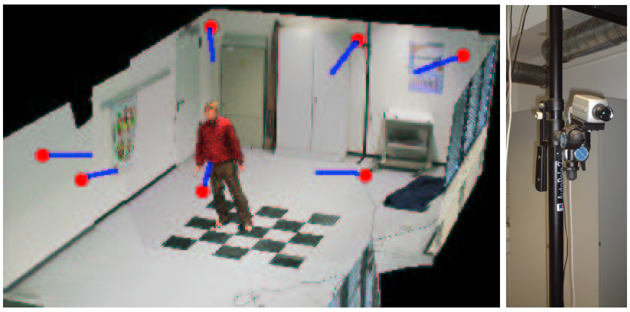


Figure 1: The figure on the left shows an example seven-camera setup used in our system. The red spheres denote camera positions and their viewing directions are shown as blue lines. The image on the right shows one of the video cameras used.

3.1 Silhouette Extraction

The inputs to the motion parameter estimation are silhouette images of the moving person from each camera perspective. The person in the foreground is separated from the background by making use of the color statistics of each background pixel (Fig. 2). From a sequence of video frames without a moving subject, the mean and standard deviation of each background pixel in each color channel are computed [Cheung et al. 2000]. If a pixel differs in at least one color channel by more than an upper threshold from the background distribution, it is classified as certainly belonging to the foreground. If its difference from the background is smaller than a lower threshold in all channels, the pixel is classified as certainly background. All other pixels are considered potential shadow pixels.

Shadows cast by the person onto the environment can easily be incorrectly classified as foreground. Image pixels in shadow show a large difference in intensity but only a small difference in hue. Exploiting this observation, shadow pixels can be identified by examining the angular difference between background and foreground hue for every pixel falling in between the lower and upper threshold. The raw silhouettes exhibit isolated noisy pixels. Silhouette quality is improved via subsequent morphological dilate and erode operations [Jain et al. 1995].

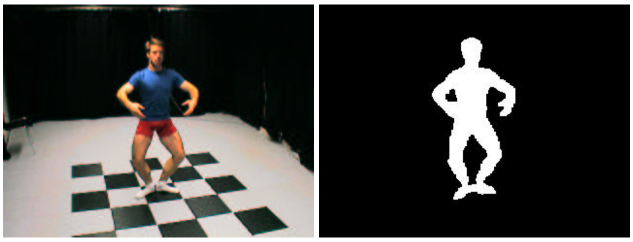


Figure 2: Comparison of an actual video frame (l) and the corresponding silhouette (r).

4 Human Body Model

The body model used throughout the system is a generic model consisting of a hierarchic arrangement of 16 body segments (head, upper arm, torso etc.). The model's kinematics are defined via an underlying skeleton consisting of 17 joints connecting bone segments. Different joint parameterizations are used in different parts of the skeleton. The root of the model located at the pelvis provides the degrees of freedom for global rotation and translation of the body. Each limb, i.e. complete arm or leg, is parameterized via four degrees of freedom. These are the position of the tip, i.e wrist or ankle, in local coordinates, and the rotation around an axis connecting root and tip (Fig. 3). This limb parameterization was chosen because it is particularly suited for an efficient grid search of its parameter space which we describe in Sect. 5.3. At every time instant, 35 parameters are need to completely define a body pose. The surface of each body segment is represented by a closed triangle mesh. To accommodate a wide range of physical body types we allow for deformation of each body segment. A separate 1D Bézier spline is defined along each coordinate axis in the local segment coordinate system which defines non-uniform scaling. Fig. 3 shows the surface model (21422 triangles) and the underlying joint structure, as well as the limb parameterization and a non-uniform scaling example.

5 Marker-free Motion Capture

The motion capture subsystem tracks the body motion of the recorded actor over time. After an initialization step, the body pose parameters that maximize the overlap between projected model silhouettes and input camera silhouettes are estimated for every time step.

5.1 Energy function

The error metric used to estimate the goodness of fit of the body model with respect to the video frames computes a pixel-wise exclusive-or between the image silhouette and the rendered model silhouette in each input camera view (Fig. 4). The energy function value is the sum of the non-zero pixels for every camera view after this pixel-wise boolean operation [Lensch et al. 2001]. This error metric is efficiently evaluated using commodity graphics hardware. At the beginning of each time step, all input camera silhouette images are transferred to the graphics card. The pixel-wise XOR is computed using the OpenGL stencil buffer. Each bit-plane of the stencil buffer is used to render the result of the overlap computation for one input camera view. The result is transferred to the main memory and the final error metric computed by summation on the CPU. Using an 8-bit stencil buffer and the depth buffer, the error metric for 8 camera views can be evaluated in graphics hardware with only a single frame buffer read and write.

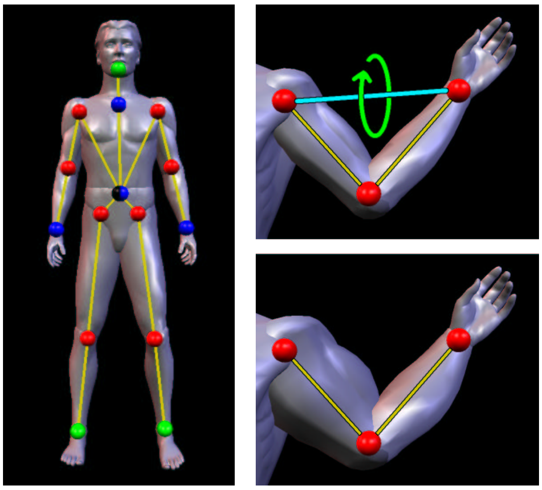


Figure 3: Surface model (l) and the underlying skeletal structure (r). Spheres indicate joints and the different parameterizations used; blue sphere - 3 DOF ball joint, green sphere - 1 DOF hinge joint, red spheres (two per limb) - 4 DOF. The black/blue sphere indicates the location of three joints, the root of the model and joints for the upper and lower half of the body. The upper right figure shows the parameterization of a limb, consisting of 3 DOF for the wrist position in local shoulder coordinates (shown in blue) and 1 DOF for rotation around the blue axis. The lower right figure demonstrates an exaggerated deformation of the arm that is possible to compute during the initialization stage.

5.2 Initialization

The motion capture is initialized using a set of silhouette images that show the human actor in an initialization pose. The ideal initialization pose is one in which both the arms and legs are bent, allowing for simple identification of elbow and knee locations. From these silhouettes, a set of scaling parameters as well as a set of pose parameters is computed. In an automatic procedure, the initial global model position is computed by using a grid sampling of the parameter space. The global model position is chosen that produces the best fit according to the error measure described in Sect. 5.1. The fit is improved by optimizing over the pose parameters and joint scaling parameters in an iterative process that employs the same error measure. In the first step of each iteration the scaling parameters of the body segments are adjusted. These include, for example, scaling parameters along bone axes for the limbs and uniform scaling parameters for the torso. The scaling is done for subparts of the body model individually using a Jacobian optimization [Press et al. 1992]. This way, the joint locations are adapted to

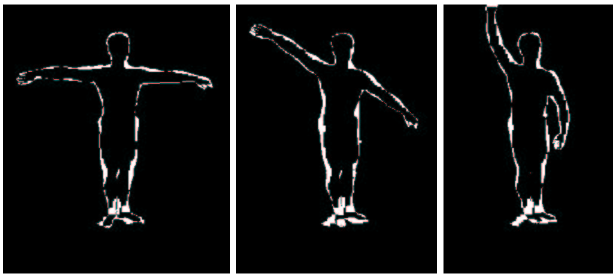


Figure 4: The energy function drives the model fitting over a series of time steps.

fit to the person's skeleton. The second step of each iteration uses the rescaled body model and computes an estimate of the body pose parameters by means of the procedure described in Sect. 5.3. These two steps are iterated several times. In a last step, the Bézier parameters (Sect. 4) controlling non-uniform scaling of segment geometry are optimized in order to adapt the model shape to the observed silhouette ever more closely. The combination of uniform and non-uniform scaling enables adaptation to a large range of body types.

After initialization, all scaling parameters are fixed, and continuous tracking is performed for all subsequent time steps. The motion parameters are computed by applying a sequence of optimizations in lower-dimensional parameter spaces.

5.3 Motion Parameter Estimation

The model parameters for each time instant are computed using a non-linear minimization approach using the previously described energy function. A straightforward approach would be to apply Powell's method [Press et al. 1992] to optimize over all degrees of freedom in the model simultaneously. This simple strategy, however, exhibits some of the fundamental pitfalls that make global optimization infeasible. In the global case, the goal function reveals many erroneous local minima. Fast movements between consecutive time frames are almost impossible to resolve correctly. For every new time step, the optimization uses the result from the previous frame as a starting point. For fast moving body segments, there will be no overlap between the starting model pose and the current time frame, and no global minimum will be found.

A different problem arises if one limb moves very close to the torso. In these cases, it is quite common for global minimization to find a local minimum in which the limb penetrates the torso.

To make the tracking procedure robust against these problems and to enable it to follow complex motions, we split the parameter estimation into a sequence of optimizations on subparts of the body.

Temporal coherence is exploited during the computation of the motion parameters. Starting from the body pose in the previous time step, the global translation and rotation of the model root are computed by using Powell's method with the energy function described in Sect. 5.1. The rotations of head and hip joints are then independently computed using an identical optimization procedure. With the main body aligned to the silhouettes, the poses of the two arms and two legs can be found with independent optimization steps. The final step in the sequence of optimizations is the computation of hand and foot orientation by optimizing over their local parameter space.

Due to the limb parameterization described in Sect. 4, fitting an arm or leg is a four-dimensional optimization problem. Considering the arm as an example, the limb fitting employs the following steps. The parameter space is efficiently constrained by applying a grid search on the four-dimensional parameter domain. The grid search samples the parameter space regularly and tests each sample for representing a valid arm pose. A valid pose is defined by two criteria. First, the wrist and the elbow must project into the image silhouettes in every camera view. Second, the elbow and the wrist must lie outside a bounding box defined around the torso segment of the model. For all detected valid poses, the error function is evaluated, and the pose possessing the minimal error is used as starting point for a downhill optimization procedure [Press et al. 1992]. The arm pose at the current time instant is the result of the downhill optimization procedure. For all 4 arm parameters (Fig. 3), the search space for valid poses is adapted to the difference in the parameter values observed during the two preceding time steps, implicitly including the assumption of a smooth arm motion into the fitting procedure.

The grid search increases the robustness of the fitting process significantly. The validity criterion for arm poses can be evaluated much faster than the error function. Energy function computations are not spent on poses which will not yield a good local minimum. By taking the best valid pose as a starting point for the final downhill minimization, the likelihood of converging to a globally optimal local minimum is significantly increased.

The overall silhouette-based motion parameter estimation has several other advantages. The algorithm is not tied to any specific body model. More complex parameterizations or different surface representations could easily be used. Furthermore, the algorithm easily scales to higher input image resolutions. Model fitting can be applied to lower resolution versions of the video frames by means of an image pyramid. On the whole, the fitting procedure exhibits a high degree of robustness and efficiency and yet is comparably simple.

6 Free-Viewpoint Rendering

After motion capture, the correct body pose is known for each time frame of the input video sequence. To create free-viewpoint video, a consistent surface texture for each time step is computed by combining the information from all available camera views. This texture is created by blending the projection of all available camera images onto the geometry [Buehler et al. 2001; Raskar and Low 2002]. In a preprocessing step, per-vertex weights are computed that indicate the influence that a specific input camera view has at a particular surface location. The precomputed weights are used during rendering to composite the body texture using projective texturing and per-pixel blending on the GPU in real-time.

Assuming that the observed human actor's body surface exhibits a Lambertian reflectance function, a consistent texture map is created for every time step. In other contexts, view-dependent texturing [Debevec et al. 1998] produces excellent results, but for non-perfectly exact object geometry our approach obtains visually significantly more pleasing results. View-dependent texturing exhibits noticeable blending artifacts in parts where the model geometry does not exactly correspond to the observed person's body shape. A time-dependent Lambertian texture produces far more visually satisfying results and preserves the high spatial frequencies for all possible viewpoints.

6.1 Texture Generation

Although our model tracking algorithm is quite robust, it is impossible for the model silhouettes to align perfectly with all input silhouettes. Some mesh vertices will project into the background in some camera views. To solve this problem, an intuitive solution would be to locally deform the model boundary to fit the silhouette boundary in all camera views. In our experiments we found that the silhouette boundaries were too noisy to make deformation on a pixel scale feasible.

As an alternative to deformation, we first remove one layer of boundary pixels in each input silhouette that potentially exhibit spatial aliasing artifacts. As a second step, each segmented input camera view is augmented in the background region by assigning to each adjacent background pixel the color value of the closest foreground pixel over a small number of iterations. This guarantees that every projected triangle spans only foreground color information.

Vertex weights are calculated based on the angle between the suface normal and viewing vector towards the input camera. For each camera, vertex visibility is determined by rendering the geometric model as a depth map and comparing the computed vertex depth with the corresponding projected value stored in the depth map. Slight misalignments between model geometry and the actual human body shape generate unsatisfactory results if the visibility calculation is not slightly modified. Texture information from occluding body parts can project onto an incorrect body segment (Fig. 6).

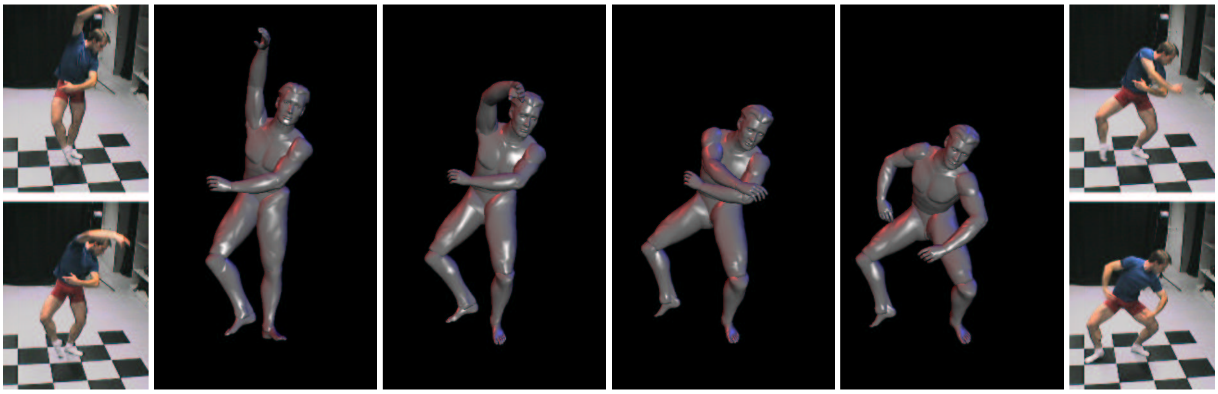


Figure 5: The four smaller images show subsequent video frames. Note the fast arm movement. In the four larger images, the corresponding poses of the body model are depicted as automatically estimated by the motion capture subsystem.

In the vicinity of occluded surface areas, another camera must be relied on to provide correct texture information. We have developed a novel approach that removes incorrect projections without resorting to computationally expensive per-pixel classification techniques.

The simple per-vertex visibility computation for each input camera is extended by additionally computing visibility from several camera views slightly displaced in the image plane. A vertex is classified as visible if and only if it is visible in both the original and displaced views.

This has the effect of removing the visibility of vertices that project into the vicinity of occluding boundaries in each camera view. Consequently, information from other camera images is used to generate a correct texture in these locations. Since a vertex is potentially visible from several camera views, per-vertex blending weights are computed for correctly compositing the surface texture.

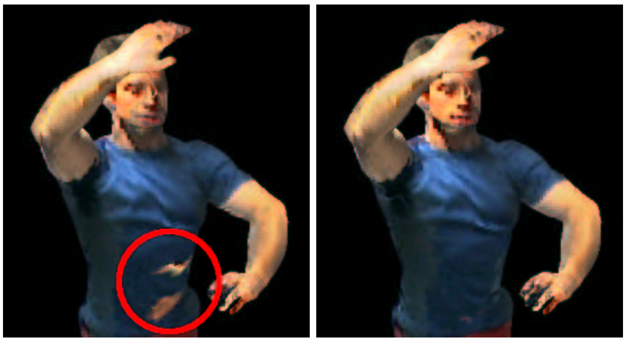


Figure 6: Incorrect texture projection (shown in red) is solved through a modified visibility calculation.

A simple way to compute these weights would be to assign to each camera view a weight defined by

$$\omega_i = \frac{1}{\Theta_i} \qquad (1)$$

where ω_i is the weight assigned to camera view i and Θ_i is the angle between the vertex normal and the viewing vector towards camera i. The drawback of this method is that fine details in the original camera views are blurred in the composite texture. To preserve fine details we employ a different weighting function to compute the per-vertex weight for each input camera $\omega_i{}'$:

$$\omega_i' = \frac{1}{(\max_i(\omega_i) + 1 - \omega_i)^{\alpha}} \qquad (2)$$

This weighting function assigns a proportionally high weight to a camera for which the angle between viewing vector and vertex normal is small. These weights are then normalized so that their sum is one. The sharpness value α controls the degree to which the largest weight is exaggerated. In the limit, as $\alpha \to \infty$, only the best camera is chosen for texture generation. Subtle details in the surface texture such as wrinkles and facial expressions are well preserved using this weighting scheme.

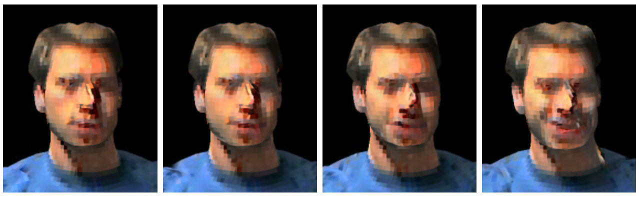


Figure 7: Capturing a smile: Texture detail is preserved. Block artifacts are due to the limited camera resolution.

6.2 Real-Time Rendering

The scene can be visualized in real-time with interactive viewpoint manipulation. At each time step, the renderer is provided with parameters of the geometric model, per-vertex texture weights, and images from the different input views. Consumer-level graphics hardware can be used to generate texture coordinates and blend the textures together based on their corresponding weights. Our viewer is implemented on NVIDIA's GeForce3$^{\text{TM}}$ rendering architecture. With four texturing units, one rendering pass is needed for each set of four cameras. The texture weights are encoded in the primary color channel and blended using the register combiner extension [Kilgard 2002].

7 Results

The proposed system has been tested on several multi-view video streams including a male ballet dancer and a second test subject (Fig. 10). Ballet dance performances are ideal test cases, as they exhibit rapid, complex motion. We chose to record sequences at a resolution of 320x240 to achieve the maximal frame rate. The motion capture subsystem demonstrates that it is capable of robustly following human motion involving fast arm motion, complex twisted poses of the extremities, and full body turns. Certainly, there are extreme body poses such as the fetal position that cannot be reliably tracked due to insufficient visibility. To our knowledge, no non-intrusive system has demonstrated it is able to track such extreme positions.

As illustrated in Fig. 5 the complex body poses are correctly recovered by the silhouette-based motion capture algorithm. An impression of the visual quality of the rendered actor can be obtained from Fig. 8. In each picture, the smaller original input images can be compared to our rendering results from the same camera perspective. This comparison shows that the original appearance of the dancer is nicely reproduced. In Fig. 9 complex body poses are textured and rendered from several novel views.

Fig. 6 depicts a problematic pose for texture generation, since in some of the input views, extremities occlude more distant parts of the body. The texture projection artifacts are effectively removed by the method described in Sect. 6.1. Subtle texture details in the clothes and the facial expression are well-preserved (Fig. 7). In our current implementation, model fitting time is dependent on the speed of the actor's motions. For slow motions, the limb parameter grid search can be confined to a narrow search space, and during optimization using Powell's method, a minimum is found very quickly. We measured fitting times for two sets of motion sequences of the male ballet dancer. In set A (dancer wears blue shirt) the motion is comparably slower to that of set B (dancer wears red "15" shirt). For set A the minimum fitting time is 1.46s with an average fitting time of 6.81s per time step. For set B the smallest fitting time is 3.46s with an average of 11.73s. Due to the efficient implementation of the energy function in graphics hardware, up to 105 energy function evaluations can be computed per second. The system scales well to a larger number of cameras. For tracking, each set of 8 cameras requires one additional memory to framebuffer read/write cycle. During rendering, on a GeForce3$^{\text{TM}}$ one pass is needed for every four cameras. Graphics chips with a higher number of texture units can significantly improve the rendering speed. We believe, though, that we have demonstrated that the presented system is capable of producing high quality results even with a comparably low number of cameras and that a higher number of cameras is not required. The free-viewpoint renderer can replay video scenes at the original captured frame rate of 15 fps. The maximal possible frame rate is significantly higher. Standard TV frame rate of 30 fps can easily be attained.

8 Discussion and Future Work

We have presented a new approach to jointly estimate and render full human body motion using a handful of synchronized video camera sequences as input. The resulting system runs on a standard PC with off-the-shelf graphics hardware and is suitable for consumer-market desktop applications. Interactive rendering frame rate, detailed body geometry, robust motion estimation, and high-quality, time-varying texture yield a realistic, natural impression of the actor. The system has been designed with free-viewpoint video as one possible application in mind. The actor's model can be set into computer-generated or recorded 3-D environments to be viewed from any arbitrary viewpoint. In addition, the estimated motion parameters can be used to animate the model of a different actor or creature.

Two major limitations currently still restrict our system. First, the motion estimation process is done off-line, making the system currently unsuitable for live broadcast applications. A faster optimization scheme that makes use of hierarchical as well as divide-and-conquer strategies may be able to sufficiently speed up the estimation step for real-time performance. One possible implementation of this idea in the future will be to distribute the optimization for separate sub-parts of the body to different GPUs. Employing this strategy and using next generation hardware, we believe that motion capture can also be achieved in real-time. Second, for the sake of high-quality texture, Lambertian reflection properties are implicitly assumed when generating one consistent texture from the input video images. While we have experimentally found this to be a valid approximation for skin and most garments on a complete-body scale, replicating view-dependent reflection effects may give an even more realistic impression. Unfortunately, object geometry must be known exactly for unblurred view-dependent texturing [Debevec et al. 1998] or estimating the Bidirectional Texture Function (BTF) [Dana et al. 1999]. By augmenting the described model-based motion capture with multi-view 3D reconstruction techniques, a reflection-consistent body surface may be determined that allows high-quality view-dependent texture mapping. Other areas of future research include making use of known lighting conditions for relighting, the extrapolation of texture for body regions that are momentarily not visible in any camera view, and the consideration of loose garments.

9 Acknowledgements

The authors would like to thank Harald Krytinar and Anna Hagermark from the Staatstheater Saarbrücken for performing in our camera room.

References

ALLEN, B., CURLESS, B., AND POPOVIC, Z. 2002. Articulated body deformations from range scan data. In *Proceedings of ACM SIGGRAPH 02*, 612–619.

BOROVIKOV, E., AND DAVIS, L. 2000. A dristibuted system for real-time volume reconstruction. In *Proceedings of Intl. Workshop on Computer Architectures for Machine Perception*, 183ff.

BOTTINO, A., AND LAURENTINI, A. 2001. A silhouette based technique for the reconstruction of human movement. *CVIU 83*, 79–95.

BREGLER, C., AND MALIK, J. 1998. Tracking people with twists and exponential maps. In *Proc. of CVPR 98*, 8–15.

BUEHLER, C., BOSSE, M., MCMILLAN, L., GORTLER, S. J., AND COHEN, M. F. 2001. Unstructured lumigraph rendering. In *Proceedings of ACM SIGGRAPH 01*, ACM Press, S. Spencer, Ed., 425–432.

CHEUNG, K., KANADE, T., BOUGUET, J.-Y., AND HOLLER, M. 2000. A real time system for robust 3D voxel reconstruction of human motions. In *Proc. of CVPR*, vol. 2, 714 – 720.

CURLESS, B., AND SEITZ, S. 2000. *3D photography Course Notes*. ACM SIGGRAPH 00.

DANA, K., VAN GINNEKEN, B., NAYAR, S., AND KOENDERINK, J. 1999. Reflectance and texture of real-world surfaces. *ACM Transactions on Graphics 18*, 1, 1–34.

DEBEVEC, P., TAYLOR, C., MALIK, J., LEVIN, G., G.BORSHUKOV, AND YU, Y. 1998. Image-based modeling and rendering of architecture with interactive photogrammetry and view-dependent texture mapping. *Proc. IEEE International Symposium on Circuits and Systems (ISCAS'98)*, Monterey, USA 5 (June), 514–517.

DELAMARRE, Q., AND FAUGERAS, O. 1999. 3D articulated models and multi-view tracking with silhouettes. In *Proc. of ICCV 99*, 716–721.

GAVRILA, D., AND DAVIS, L. 1996. 3D model-based tracking of humans in action: A multi-view approach. In *Proc. of CVPR 96*, 73–80.

GAVRILA, D. 1999. The visual analysis of human movement. *CVIU 73*, 1 (January), 82–98.

GRAMMALIDIS, N., GOUSSIS, G., TROUFAKOS, G., AND STRINTZIS, M. 2001. Estimating body animation parameters from depth images using analysis by synthesis. In *Proc. of Second International Workshop on Digital and Computational Video (DCV'01)*, 93ff.

JAIN, R., KASTURI, R., AND SCHUNCK, B. 1995. *Machine Vision*. McGraw-Hill.

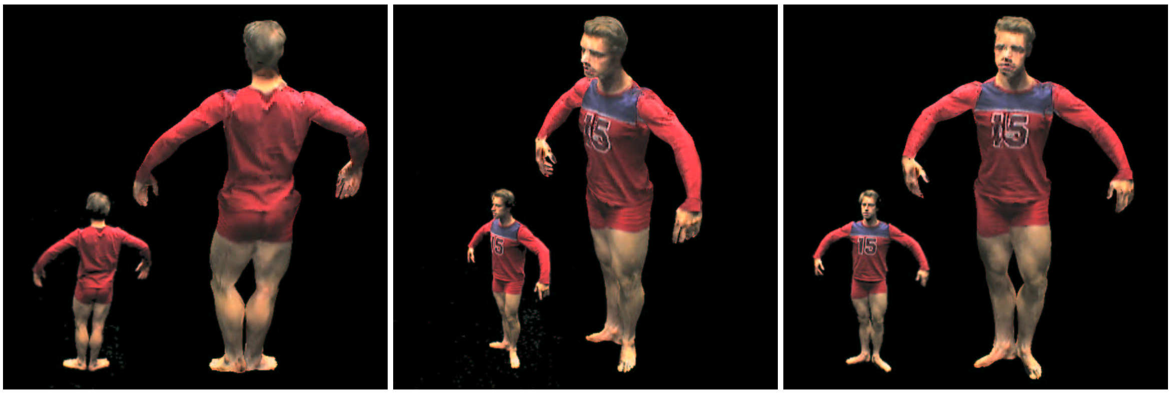


Figure 8: For comparison, segmented input images (small) and the resulting rendered views corresponding to the same perspective (large) are depicted.

KILGARD, M. J., 2002. Nvidia opengl extension specifications. http://developer.nvidia.com/docs/IO/3260/ATT/nv30specs.pdf.

KOENEN, R., 2002. Mpeg-4 overview. http://mpeg.telecomitalialab.com/standards/mpeg-4/mpeg-4.htm.

LAURENTINI, A. 1994. The visual hull concept for silhouette-based image understanding. *Pattern Analysis and Machine Intelligence 16*, 2 (February), 150–162.

LENSCH, H., HEIDRICH, W., AND SEIDEL, H. P. 2001. A silhouette-based algorithm for texture registration and stitching. *Graphical Models 64(3)*, 245–262.

LEUNG, M., AND YANG, Y. 1995. First sight : A human body outline labeling system. *PAMI 17(4)*, 359–379.

LEVOY, M., AND HANRAHAN, P. 1996. Light field rendering. In *Proceedings of ACM SIGGRAPH 96*, vol. 30, 31–42.

LUCK, J., AND SMALL, D. 2002. Real-time markerless motion tracking using linked kinematic chains. In *Proc. of CVPRIP02*.

MARTINEZ, G. 1995. 3D motion estimation of articulated objects for object-based analysis-synthesis coding (OBASC). In *VLBV 95*.

MATSUYAMA, T., AND TAKAI, T. 2002. Generation, visualization, and editing of 3D video. In *Proc. of 1st International Symposium on 3D Data Processing Visualization and Transmission (3DPVT'02)*, 234ff.

MATUSIK, W., BUEHLER, C., RASKAR, R., GORTLER, S., AND MCMILLAN, L. 2000. Image-based visual hulls. In *Proceedings of ACM SIGGRAPH 00*, 369–374.

MATUSIK, W., BUEHLER, C., AND MCMILLAN, L. 2001. Polyhedral visual hulls for real-time rendering. In *Proceedings of 12th Eurographics Workshop on Rendering*, 116–126.

MENACHE, A. 1995. *Understanding Motion Capture for Computer Animation and Video Games*. Morgan Kaufmann.

MIKIĆ, I., TRIVERDI, M., HUNTER, E., AND COSMAN, P. 2001. Articulated body posture estimation from multicamera voxel data. In *Proc. of CVPR*.

MOEZZI, S., TAI, L.-C., AND GERARD, P. 1997. Virtual view generation for 3D digital video. *IEEE MultiMedia 4*, 1 (Jan.–Mar.), 18–26.

MULLIGAN, J., AND DANIILIDIS, K. 2000. View-independent scene acquisition for telepresence. In *Proceedings of the International Symposium on Augmented Reality*, 105–108.

NARAYANAN, P., RANDER, P., AND KANADE, T. 1998. Constructing virtual worlds using dense stereo. In *Proc. of ICCV 98*, 3 – 10.

PLAENKERS, R., AND FUA, P. 2001. Tracking and modeling people in video sequences. *CVIU 81*, 3 (March), 285–302.

PRESS, W., TEUKOLSKY, S., VETTERLING, W., AND FLANNERY, B. 1992. *Numerical Recipes*. Cambridge University Press.

RASKAR, R., AND LOW, K.-L. 2002. Blending multiple views. In *Proceedings of Pacific Graphics 2002*, 145–153.

ROHR, K. 1993. Incremental recognition of pedestrians from image sequences. In *Proc. of CVPR 93*, 8–13.

SILAGHI, M.-C., PLAENKERS, R., BOULIC, R., FUA, P., AND THALMANN, D. 1998. Local and global skeleton fitting techniques for optical motion capture. In *Modeling and Motion Capture Techniques for Virtual Environments*, Springer, no. 1537 in LNAI, No1537, 26–40.

TERZOPOULOS, D., CARLBOM, I., FREEMAN, W., KLINKER, G., LORENSEN, W., SZELISKI, R., AND WATERS, K. 1995. Computer vision for computer graphics. In *ACM SIGGRAPH 95 Course Notes*, vol. 25.

THEOBALT, C., MAGNOR, M., SCHUELER, P., AND SEIDEL, H.-P. 2002. Combining 2D feature tracking and volume reconstruction for online video-based human motion capture. In *Proceedings of Pacific Graphics 2002*, 96–103.

TSAI, R. 1986. An efficient and accurate camera calibration technique for 3D machine vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'86)*, 364–374.

VEDULA, S., BAKER, S., AND KANADE, T. 2002. Spatio-temporal view interpolation. In *Proceedings of the 13th ACM Eurographics Workshop on Rendering*, 65–75.

WREN, C., AZARBAYEJANI, A., DARRELL, T., AND PENTLAND, A. 1997. Pfinder: Real-time tracking of the human body. *PAMI 19*, 7, 780–785.

WUERMLIN, S., LAMBORAY, E., STAADT, O., AND GROSS, M. 2002. 3d video recorder. In *Proceedings of Pacific Graphics 2002, IEEE Computer Society Press*, 325–334.

YONEMOTO, S., ARITA, D., AND TANIGUCHI, R. 2000. Real-time human motion analysis and IK-based human figure control. In *Proceedings of IEEE Workshop on Human Motion*, 149–154. .

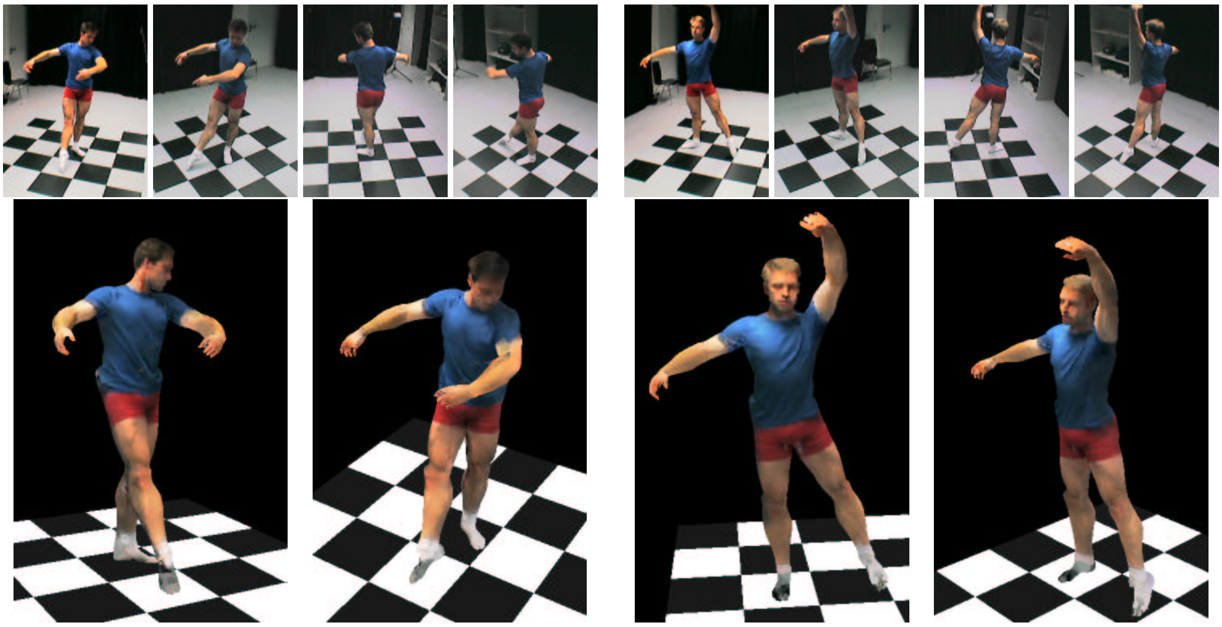


Figure 9: Novel viewpoints are realistically synthesized. Two distinct time instants are shown on the left and right with input images above and novel views below.

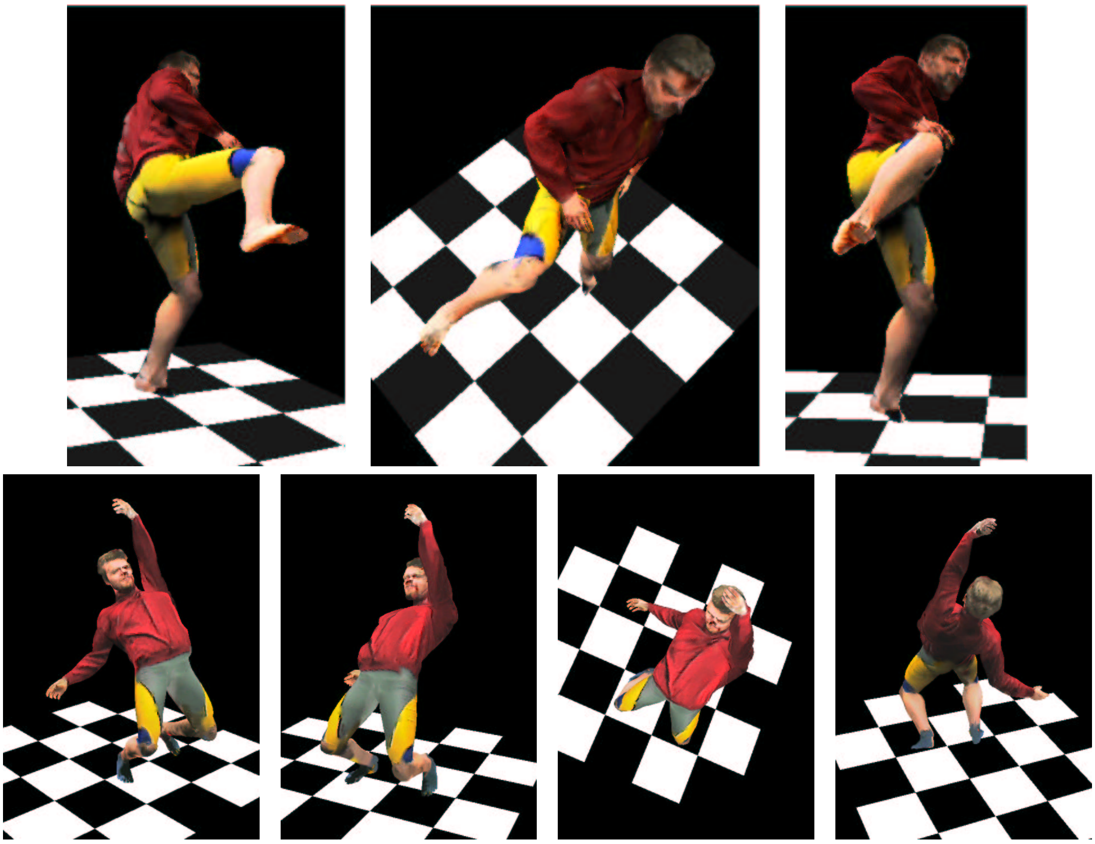


Figure 10: Free-viewpoint video allows the viewer to experience karate kicks in a whole new light (top row). Conventional video systems cannot offer moving viewpoints of scenes frozen in time. However, with our free-viewpoint video system *freeze-and-rotate* camera shots of instable body poses are possible (bottom row).