Optical Motion Capture: Theory and Implementation

Gutemberg B. Guerra-Filho1

Abstract: Motion capture is the process of recording real life movement of a subject as sequences of Cartesian coordinates in 3D space. Optical motion capture (OMC) uses cameras to reconstruct the body posture of the performer. One approach employs a set of multiple synchronized cameras to capture markers placed in strategic locations on the body. A motion capture system has applications in computer graphics for character animation, in virtual reality for human control-interface, and in video games for realistic simulation of human motion. In this tutorial, we discuss the theoretical and empirical aspects of an optical motion capture system. Basically, for a motion capture system implementation; the resources required consist of a number of synchronized cameras, an image acquisition system, a capturing area, and a special suit with markers. The locations of the markers on the suit are designed such that the required body parts (e.g. joints) are covered. We present our motion capture system using a framework that identifies different sub-problems to be solved in a modular way. The sub-problems involved in OMC are initialization, marker detection, spatial correspondence, temporal correspondence, and post-processing. In this tutorial, we discuss the theory involved in each sub-problem and the corresponding novel techniques used in the current implementation. The initialization includes setting up a human model and the computation of intrinsic and extrinsic camera calibration. Marker detection involves finding

1 Computer Vision Laboratory, Center for Automation Research.
University of Maryland, College Park, MD 20742-3275.
{guerra@cs.umd.edu}

the 2D pixel coordinates of markers in the images. The spatial correspondence problem consists in finding pairs of detected markers in different images captured at the same time with different viewpoints such that each pair corresponds to the projections of the same scene point. Given camera calibration and the spatial matching, the 3D reconstruction of markers (translational data) is achieved by triangulating the various camera views. The temporal correspondence problem (tracking) involves matching two clouds of 3D points representing detected markers at two consecutive frames, respectively. The temporal correspondence module builds a track for each marker where the marker's 3D coordinates are concatenated according to time. Post-processing consists in labeling each track with a marker code, finding missing markers lost by occlusions, correcting possible gross errors, and filtering noise. Once the translational data is processed, a hierarchical human model may be used to compute rotational data (joint angles). We consider standard data formats available for motion capture data (e.g. bvh, acclaim). Other important techniques used to improve consistency in the motion data are volumetric reconstruction, inverse kinematics, and inverse dynamics. We also cover topics related to editing and manipulation of motion data.

1 Introduction

Optical motion capture (OMC) is an important field in computer vision widely used in computer graphics and responsible for advances in many research areas. The importance of OMC is mostly due to the relevant problems involved in the process and to the numerous applications for real motion data. Realistic movement is required to perform synthesis and analysis of human motion. Motion synthesis consists in simulate, control, or create new object/subject movement. In synthesis, motion capture data improves believability of human rendering and brings personality to animated characters. In motion analysis, the captured information is used to evaluate some aspects of the musculo-skeletal system. An optical motion capture system is a convenient means for extracting detailed information from a subject in order to track its movement.

Motion capture systems can be divided into magnetic, mechanic, and optical. Magnetic systems use electromagnetic sensors connected to a computer which can produce 3D data in real-time with low processing costs. However, a magnetic system restricts movement due to cabling. Mechanical systems use special suits with integrated mechanical sensors that register the motion of articulation in real-time and with no processing. Optical systems are based on photogrammetric methods. Optical systems provide high accuracy, complete freedom of movement, and the possibility of interaction between different actors with a higher computational cost.

Optical motion capture is further classified according to the number of views (monocular or multi-view) and to the use of markers (marker-based or markerless). Monocular systems use images acquired by a single camera, while multi-view systems use images acquired simultaneously and synchronously by two or more cameras. Monocular techniques have to deal with ambiguities in the reconstruction of the 3D pose caused by reflective ambiguity [25] and kinematic singularities [21]. A marker-based system measures the trajectories of target points (markers) on the body, while markerless systems compute motion parameters from extracted silhouettes or other features (e.g. edges). In this tutorial, we will cover the theory and implementation of a multi-view marker-based optical motion capture system that we have developed in our laboratory.

Among the various research areas where OMC has applications, we find computer graphics for the animation of visually convincing human characters. In visual media, computer-animated human characters are widely used in movie productions and television commercials. Human motion capture is used to acquire motion data from a video of a real moving person. Subtle gestures are recorded to convey emotion through motion. Applications include virtual reality for human control-interface and video games for realistic simulation of human motion. Optical motion capture is also effective in a clinical context for medical investigation such as the assessment of orthopedic pathologies. Other research areas are kinesiology and biomechanics for movement analysis, performance and injuries research

in sports; dance for annotation when data is translated into dance notation systems; and robotics for robots control.

The optical motion capture topic involves many non-trivial problems in computer vision. This tutorial covers both theoretical and empirical aspects of OMC and offers knowledge on computer vision and about a technique widely used in computer graphics. The important topics on computer vision discussed in this tutorial include camera calibration, feature detection, stereo matching, tracking, and volumetric reconstruction. The covered topics on computer graphics are inverse kinematics, inverse dynamics, editing, and manipulation of motion capture data.

2 Required Resources

In optical motion capture, multi-view image data is used to compute the time-varying motion parameters of a moving person. Multi-view image data consists in synchronized video streams of dynamic scenes recorded from multiple cameras. The resources required to acquire multi-view image data involve a capture room, body suit, camera equipment, and an acquisition system. Our optical motion capture system uses the Keck Lab facilities at the University of Maryland [11] and the acquisition system is the Argus Eye [22].

2.1 Capture Room

The spatial *dimensions* of the capture room need to be large enough to allow recording from a large number of viewpoints at sufficient distance. Opaque curtains may surround the capture area in order to improve background subtraction. Uniform *lighting* is an important issue concerning the minimization of sharp shadows cast on the floor and unwanted highlights on the scene. Sufficient illumination is achieved with fixed light sources and reflectors that spread the light homogeneously. Direct recording of the light sources should be avoided since it would cause glares in the camera optics. A separated area is dedicated as a *control room*. The control room must be shielded from view of the cameras.

2.2 Body Suit

In an optical motion capture system, a moving person (actor) wears a special body suit with markers on some body parts. The main properties of a marker are position, color, and shape (e.g. spherical or rectangular). The markers need to be placed in locations where the skin is close to the bone in order to avoid skin sliding. In general, markers are attached to specific joint positions or feature points of the human body. However, joint location may be computed indirectly from markers located at general places in the same rigid body part. If the colors of the markers are well distinguished from one another, the marker detection is more robust.

The MoCap suit consists of a cap (hood or mask are alternatives), a one-piece tight-fitting leotard with long sleeves, gloves, and socks. Ideally, the colors of the suit, markers, and background should be distinct. However, in a MoCap system using grayscale cameras, image intensity has a large variability for a fixed color due to shadows and orientation changes. In this case, either the body suit or the markers should have the same color as the background. A different color for the body suit will make silhouette segmentation easier, while a distinct color for the markers will facilitate marker detection. The subject of our capture has 49 white rectangular markers attached to black body suit contrasted with a white background.

2.3 Camera Equipment

Progressive scan cameras acquire full frames at the same time while interlaced cameras divides a full frame into two fields (odd and even lines) recorded consecutively at different times. In order to avoid a saw pattern in motion video, progressive scan cameras are preferred. The cameras must allow *synchronization* among the other cameras in order to register the multiple video streams correctly in time. External synchronization is possible via a trigger pulse sent from the parallel port of a computer to each camera. The signal is transmitted via coaxial cables to an external connector in each camera. The cameras have to deliver high *frame rates* at a suitable *resolution*. Image resolution is related to the accuracy in the determination of a feature's 3D position through triangulation. Motion blur is avoided by reducing the camera exposure time while camera focus should also be set to avoid defocus effects. The frames are digitally transferred to the acquisition system using *bus cables* (USB or FireWire). The cameras are positioned in telescope tripods or poles with 3 degree-of-freedom mounting brackets. Lenses are selected according to field of view and lens distortion. Usually, lens distortion increases with field of view. Lenses with significant image defocus should be avoided since it cannot be corrected in software as radial distortion. The multi-view vision system we use has eight synchronized cameras in a circular arrangement.

2.4 Acquisition System

The acquisition system must be able to acquire synchronous video data in real-time, to provide bandwidth for recording, and storage for saving multiple video streams. For multiple cameras, the transfer of the image data to the computer needs to be efficient and requires one bus (USB or FireWire) connector for each camera. A sufficient I/O bandwidth has to be available in the computer bus system.

Software is required for trigger control and to interface the hardware components (control of the cameras). Camera interface involves asynchronous reading of camera frames and visualization of images on the screen. A grabber system transfers captured images to memory and writes them to the disks. Table 1 presents the configurations of three vision laboratories, which could be applied to motion capture.

Table 1. Optical Motion Capture Requirements.

	Keck Lab [11]	Argus Eye [22]	VideoRoom [29]
Room Dimensions	7m x 7m x 3m	7m x 7m x 3m	11m x 5m x 3m
Camera	Kodak^{TM} ES-310	Sony^{TM} XCD-X700	Sony^{TM} DFW-V500
Camera Number	64	9	8
Resolution (pixels)	648 x 484	1024 x 768	640 x 480
Frame Rate (fps)	85	15	30

3 Markers Configuration

A marker configuration should cover all the major joints. An optical motion capture system records translational data for individual points corresponding to markers. The marker setup should be able to capture internal rotations of the main joints, including at least four sections of the spine. Sources for anthropomorphic feature points of the human body are found in the h-anim project for humanoid animation [1] and in the CAESAR project: Civilian American and European Surface Anthropometry Resource Project [2]. Menache [20] describes the design of a marker setup that allows joint rotation computation.

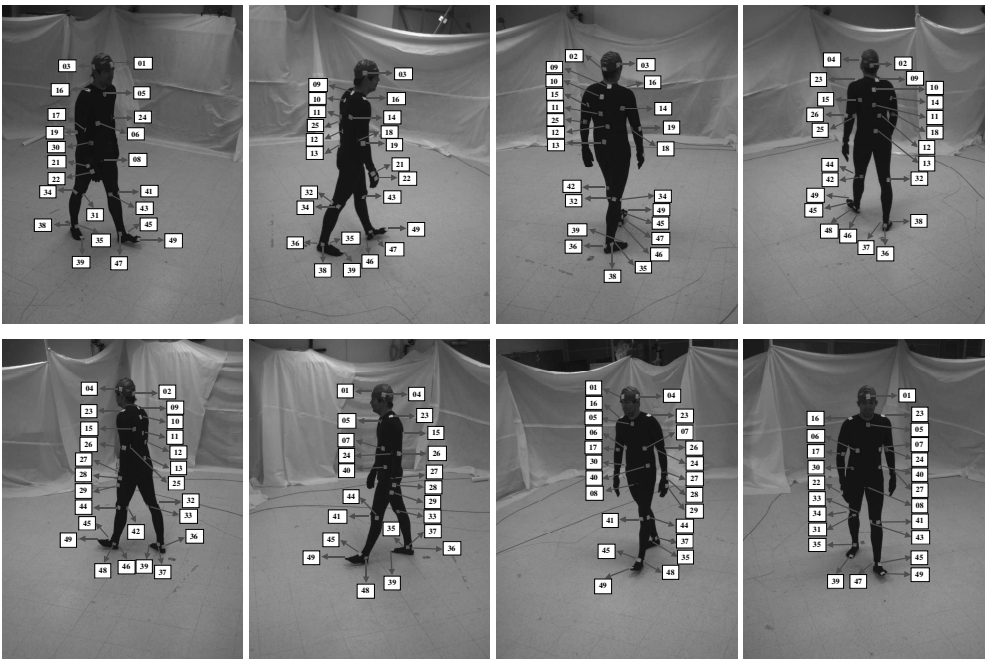


Fig. 1: The MoCap body suit with white rectangular markers.

Our marker setup covers rigid segments including pelvis, vertebral column, neck, head, shoulders, upper and lower arms, hands, upper and lower legs, and feet (see Fig. 1). A

total of 49 white rectangular markers are placed on a black body suit. Table 2 lists the set of markers and the corresponding numerical codes, code names, and attachment location.

Table 2. The set of markers in our configuration.

Code	Name	Location
01	FOREHEAD	front of the head at the middle of the forehead;
02	NUCHALE	back of the head at the nuchale;
03	R_HEAD	right side of the head;
04	L_HEAD	left side of the head;
05	STERNUM	bottom of the sternum in the chest;
06	R_RIB	right rib 10;
07	L_RIB	left rib 10;
08	CROTCH	above the genitalia;
09	T1	behind the T1 vertebra in the thoracic spine;
10	T4	behind the T4 vertebra in the thoracic spine;
11	T7	behind the T7 vertebra in the thoracic spine;
12	L1	behind the L1 vertebra in the lumbar spine;
13	SACRUM	middle of the sacrum in the back;
14	R_AXILLA	right axilla in the posterior side at the top of the right scapula;
15	L_AXILLA	left axilla in the posterior side at the top of the left scapula;
16	R_SHOULDER	above the right acromion in the shoulder girdle;
17	RF_ELBOW	around the right elbow at the arm fold;
18	RI_ELBOW	right medial humeral epicondyle, interior side of the elbow;
19	RE_ELBOW	right lateral humeral epicondyle, exterior side of the elbow;
20	RR_WRIST	right radial styloid in the wrist;
21	RU_WRIST	right ulnar styloid in the wrist;
22	R_HAND	above the metacarpals (middle of outer part) in the right hand;
23	L_SHOULDER	above the left acromion in the shoulder girdle;
24	LF_ELBOW	around the left elbow at the arm fold;
25	LI_ELBOW	left medial humeral epicondyle, interior side of the elbow;
26	LE_ELBOW	left lateral humeral epicondyle, exterior side of the elbow;
27	LR_WRIST	left radial styloid in the wrist;
28	LU_WRIST	left ulnar styloid in the wrist;
29	L_HAND	above the metacarpals (middle of outer part) in the left hand;
30	R_HIP	right superior head of the femur;
31	RF_KNEE	front of the right knee;
32	RB_KNEE	back of the right knee;
33	RI_KNEE	right femoral medial epicondyle at the interior side of the knee;
34	RE_KNEE	right femoral lateral epicondyle at the exterior side of the knee;
35	RF_ANKLE	front of the right ankle;
36	RB_ANKLE	back of the right ankle at the heel;
37	RI_ANKLE	right medial malleolus at the interior side of the ankle;
38	RE_ANKLE	right lateral malleolus at the exterior side of the ankle;
39	R_TOE	right metatarsal phalange of the great toe;
40	L_HIP	left superior head of the femur;
41	LF_KNEE	front of the left knee;
42	LB_KNEE	back of the left knee;
43	LI_KNEE	left femoral medial epicondyle at the interior side of the knee;
44	LE_KNEE	left femoral lateral epicondyle at the exterior side of the knee;
45	LF_ANKLE	front of the left ankle;
46	LB_ANKLE	back of the left ankle at the heel;
47	LI_ANKLE	left medial malleolus at the interior side of the ankle;
48	LE_ANKLE	left lateral malleolus at the exterior side of the ankle;
49	L_TOE	left metatarsal phalange of the great toe;

Many markers have a self-explaining code name, but a detailed description of the attachment location of each marker is useful. The vertebral column is divided into lumbar spine, thoracic spine, and cervical spine. The lumbar spine corresponds to one rigid segment associated with markers at the L1 vertebra and SACRUM. The thoracic spine is divided into three segments: upper, middle, and lower thorax. The lower thoracic segment is associated with markers at the T7 and L1 vertebrae. The middle thoracic segment is defined by markers at the T4 and T7 vertebrae. The upper thoracic segment is related to markers at the T1 and T4 vertebrae. The cervical spine (neck) is modeled as one segment from the T1 vertebra to the NUCHALE. The same subset of markers in the right arm (leg) is used symmetrically in the left arm (leg). At least four points are needed to define the motion of the forearm. The lower arm quadrangle contains markers in the elbow and wrist. The hand and foot are captured as a single rigid segment.

4 Initialization

The initialization consists in the calibration of many aspects of the optical motion capture system. The initial setup requires camera calibration to find intrinsic and extrinsic camera parameters, metric calibration to align the world coordinate system, background calibration to perform the special case of motion segmentation, and skeleton calibration to build a kinematic human body model.

4.1 Camera Calibration

Assuming the pinhole camera model, a 3D point X_j is projected into a 2D image point ${u_j}^i$ according to $\lambda_j^i {u_j}^i = P^i X_j$, where $\lambda_j^i \in \Re^+$ and P^i is a 3×4 projection matrix defined by 11 camera parameters. The position and orientation of the camera are described by six parameters (called extrinsic or external) while the camera properties are specified by five parameters (called intrinsic or internal).

Usually, the intrinsic and extrinsic camera parameters are computed by a calibration procedure which uses a reference object (see Fig. 2). Tsai's algorithm [30, 31] estimates the external and internal parameters using a checkerboard pattern attached to a hard panel. A video where the checkerboard pattern covers a large part of each camera's field of view is given to the calibration procedure. However, when the calibration device is a moving plane, it is not visible in all cameras and the partial calibrations have to be chained in a process prone to errors.

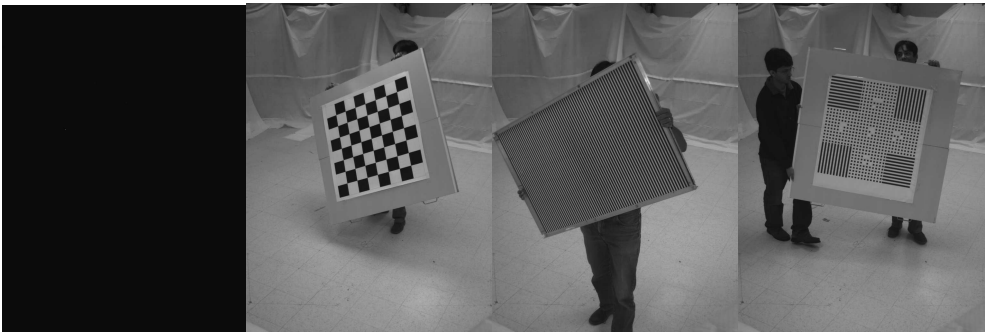


Fig. 2: Calibration devices used in the Keck Lab for the Argus Eye.

In order to have a calibration device visible from all cameras, Svoboda [3, 28] proposes a calibration method, which uses light points from laser pointers or LEDs. In a calibration video, the light point is waved throughout the working volume. The light points are detected independently in each camera and a 2D Gaussian is used to reach sub-pixel precision. These points are merged over time to represent the images of a virtual 3D calibration device. The points are validated through a pairwise epipolar constraint using a RANSAC 7-point algorithm [14].

Given pixel coordinates u_j^i of n points ($j = 1, \ldots, n$) observed from m cameras ($i = 1, \ldots, m$), a calibration procedure estimates the scales λ_j^i and the projection matrices P^i. The *scaled measurement matrix* $W_{3m \times n}$ is constructed with all image 2D observed points u_j^i, the *projective motion matrix* $P_{3m \times 4}$ represents all camera projections P^i, and the *projective shape matrix* X represents all 3D points. Computing the scales λ_j^i with a factorization algorithm [27], then matrix W = PX has rank 4 and can be factorized into P and X.

The factorization recovers motion and shape up to a homography $H_{4 \times 4}$ such that W = PX = P'X', where P' = PH and X' = H^{-1}X. The self-calibration process computes a matrix H such that P' and X' become Euclidean [16]. The Euclidean stratification finds the appropriate H by imposing geometrical constraints (e.g. orthogonality of rows and columns). The minimal number of cameras depends on the number of camera parameters known or unknown but same for all cameras: counting argument.

Radial distortion is removed by giving the observed image 2D points and the 3D reconstructed points to a standard method of nonlinear distortion estimation [4] and performing self-calibration with the undistorted points. The parameters of the non-linear distortion are computed through iterative refinement.

4.2 World Coordinate System Alignment

The calibration procedure yields the external camera parameters in some general coordinate system. A world coordinate system with the Z plane coincident with the ground floor must be aligned with the cameras coordinate system. In order to perform this

alignment, scene points with known physical dimensions and positions must be localized in the camera images.

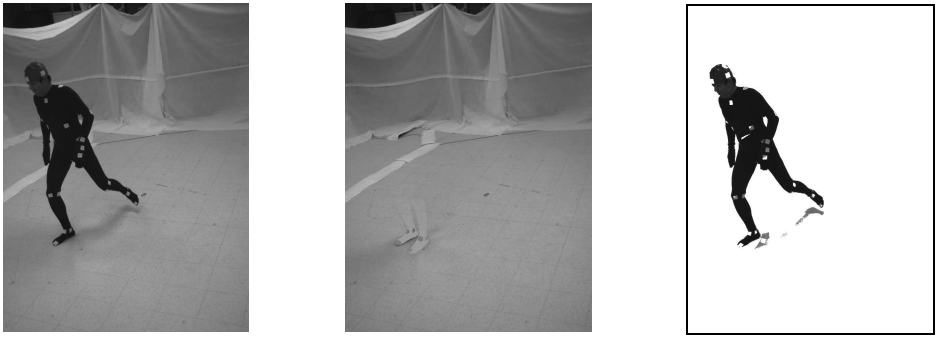
In our alignment process, a user manually selects a set of alignment points p_i, (i = 1, …, 5) in the ground floor for each camera view, where p_0 is the origin of the world coordinate system and $\{p_1, p_2, p_3, p_4\}$ are the middle points of the segments corresponding to the sides of a square centered at p_0 with size length of 2ft (60.96cm). Using camera calibration, the alignment points are reconstructed as 3D points in the camera coordinate system. The average Euclidean distance d from p_0 to the other alignment points corresponds to 1ft in real world and is used to find a metric scale.

A similarity transformation is computed with these dimensions and positions [6]. This transformation M corresponds to a translation T and a rotation R such that $M = RT$. Let v_1 be the vector $\overrightarrow{p_1 p_3}$, v_2 be the vector $\overrightarrow{p_2 p_4}$, and $\hat{v}_3$ be the cross product of the normalized vectors $\hat{v}_1$ and $\hat{v}_2$; then the rotation and translation are defined as:

$$R = \begin{bmatrix} -\hat{v}_{1x} & -\hat{v}_{1y} & -\hat{v}_{1z} & 0 \\ -\hat{v}_{2x} & -\hat{v}_{2y} & -\hat{v}_{2z} & 0 \\ -\hat{v}_{3x} & -\hat{v}_{3y} & -\hat{v}_{3z} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ and } T = \begin{bmatrix} 1 & 0 & 0 & -p_{0x} \\ 0 & 1 & 0 & -p_{0y} \\ 0 & 0 & 1 & -p_{0z} \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

4.3 Background Subtraction

Robust separation of the foreground object from the background is essential to avoid objects whose colors are similar to those of the color markers. If a background with a uniform color (blue or green) is available, a chroma-keying approach may be used [32]. Background video from each viewpoint is captured under the same lighting condition of motion capture. The mean color and standard deviation of each background pixel is computed from the background video: a sequence of images without any foreground. The foreground is identified by a large deviation of color (intensity) from the background statistics (see Fig. 3). Shadows are characterized by a large intensity difference but only a small hue difference [10].



(a) Original image. (b) Median background. (c) Subtracted image.

Fig. 3: Background subtraction.

4.4 Kinematic Human Body Model

A dynamic model considers information related to forces and torques and, consequently, requires the weight of body parts. A kinematic model for an articulated body concerns with angular velocity and acceleration of joints. The information related to a kinematic model includes the geometry and topology of a skeleton.

An articulated human body model (virtual skeleton) consists of a number of connected subparts (bony segments) subject to complex motion, where each subpart is assumed to be rigid and linked to other subparts through joints (see Fig. 4). A joint is the intersection of two segments. A human body model includes a skeleton topology and fixed properties such as link lengths and axes of rotation.

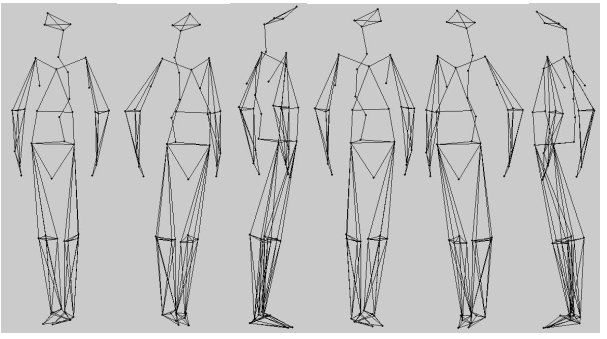


Fig. 4: The rigid segments of an articulated human body model.

A model topology consists in the connectivity between adjacent links. The geometric size of each subpart must be given to the process that estimates body pose. The pose or posture of the kinematic model is the spatial state (position and orientation) of each subpart with respect to the world coordinate system. The state does not include the lengths of the

links, since they are known and assumed constant throughout the motion (rigidity assumption) according to the model. Empirically the distance between two markers is not constant, but contained within an interval. Therefore, the statistics of the segments is computed to define boundaries to the variation in length of each link. The statistics is computed from a variety of movements performed by the same subject wearing the same markers and results in a skeleton scaled to the body proportions of the performing subject. Anthropometric tables may be used to extrapolate de model and infer the length of other skeleton segments for the subject.

5 Marker Detection

In our motion capture system, features on the body suit are detected as 2D points in each image frame. The locations of the identified markers on each camera image provide the input to the motion estimation process. These points are due to either a passive marker placed on the body suit or an error in the detection process. In commercial systems, retroreflective markers are detected by special dedicated hardware (e.g. infrared cameras) in real-time [8]. The images are filtered with a high pass filter and thresholding is performed. Our optical motion capture system implements three marker detectors based on intensity segmentation, perimeter curvature, and edges. Ideally, the system returns a 2D point location for each visible marker and each camera that sees it.

In our intensity segmentation based marker detector, the foreground is computed using background subtraction and further segmented into the body suit (dark) and the markers (bright) using thresholding (see Fig. 5). The segmentation results in a binary image for the markers imaged as bright pixels. The connected components for the bright pixels represent marker candidates. A connected component is detected as a marker if its area size is between some short interval. The candidate marker point in the image is the centroid point of the corresponding connected component.

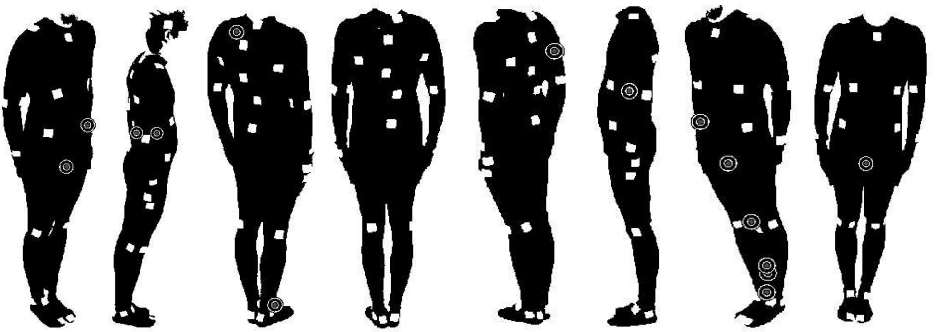


Fig. 5: Marker detector using binary thresholding.

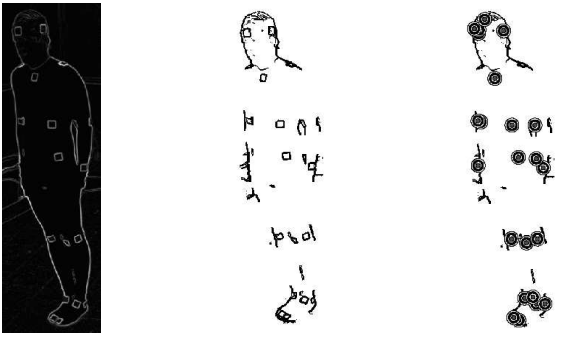
A second marker detector is based on the curvature of the dark body suit perimeter. The body suit (dark pixels) is segmented in the foreground. The perimeter of the resulting

binary image is computed (see Fig. 6). The pixels in the perimeter with a curvature above some limit are considered corners. Every pair of corner pixels defines a line segment which is possibly over a marker if the segment is fully contained in the bright region. The middle point of these segments serves as the center of a ball region which defines center pixels. The connected components for the center pixels are detected as markers. The marker location is the centroid point of the corresponding connected component.

(a) Perimeter. (b) Corners. (c) Centers. (d) Markers.

Fig. 6: Marker detector using curvature.

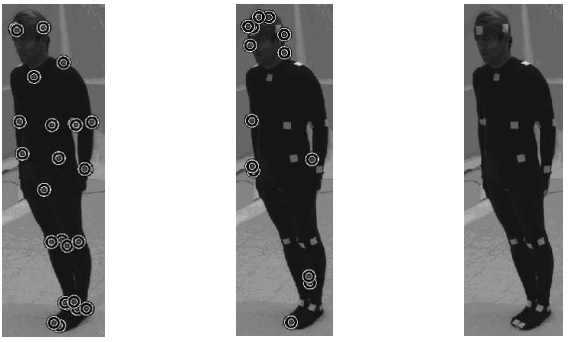
Another marker detector implemented takes advantage of edge computation from varying gradient magnitude thresholding The image gradient magnitude is defined as the length of the vector with components as the horizontal and vertical spatial derivatives of the image (see Fig. 7). The edges of the image are computed for a range of thresholds applied to the gradient magnitude. The edge image for a particular threshold value is called a partial edge image. The final edge pixels are those with cumulative participation in a certain number of partial edge images. The connected components for the final edges are detected as markers.



(a) Gradient. (b) Final edges. (c) Markers.

Fig. 7: Marker detector using edges.

Detection errors lead to detected points that do not correspond to any markers in the setup: a false detection (see Fig. 8). Not every marker is detected in each frame and camera. A particular marker may not be detected in every camera, either because it was occluded from some camera view or because the detection process failed.



(a) True detection. (b) False detection. (c) False negatives.

Fig. 8: Results for all marker detectors.

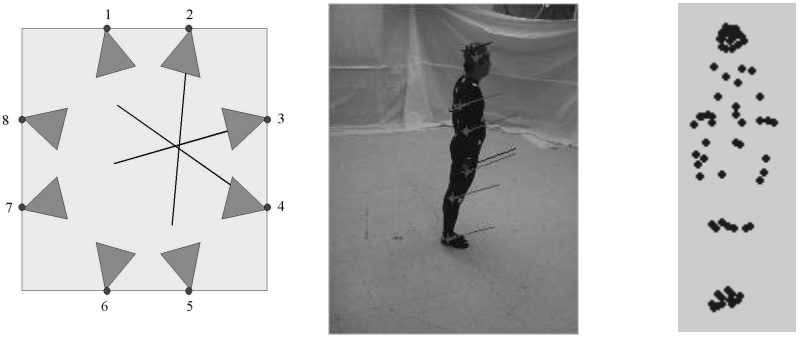
6 Spatial Correspondence

The spatial correspondence concerns in establishing matches between markers detected in images of different views. If a marker is detected on an image, its 2D point location is determined in the image plane. When these multiple sets of 2D points (one for each camera) are matched, they are reconstructed into 3D space according to a multi-colinearity constraint.

Given the coordinates of a marker in at least two different camera image planes, stereo triangulation on rays (forward intersection) is used to compute the 3D position of this marker in the scene [13]. Using calibration (projection matrix) for each camera, a detected marker corresponds to a ray originated at the camera center and passing through the marker's 2D location. The intersection point of rays corresponding to the same marker from multiple cameras is the position of the marker in 3D space. In practice, the rays do not intersect precisely due to noise (see Fig. 9).

The linear reconstruction of 3D points from multiple perspective views takes as input the homogeneous coordinates of observed 2D points and the projection matrices of the respective camera views. A 3D point $X_{4\times1}$ is projected into an observed 2D point $x_{3\times1}$ according to the equation $x = PX$, where $P_{3\times4}$ is a camera matrix. The projective equation is equivalent to a vector cross product $x \times PX = 0$, where $x = [x_1\ x_2\ 1]^T$ and $PX = [P_1{}^TX\ P_2{}^TX\ P_3{}^TX]^T$. The cross product expands only to two independent equations: $x_2(P_3{}^TX) - (P_2{}^TX) = 0$ and $x_1(P_3{}^TX) - (P_1{}^TX) = 0$. For all cameras, the independent equations $(x_2P_3{}^T - P_2{}^T)X = 0$ and $(x_1P_3{}^T - P_1{}^T)X = 0$ are combined into a homogeneous system with the form $AX = 0$. If UDV^T

is the singular value decomposition of matrix A, the 3D reconstructed point X is the last column of the V matrix.



(a) Stereo triangulation. (b) Disparity vectors. (c) Cloud of 3D points.

Fig. 9: The spatial correspondence using camera calibration.

The epipolar constraint may be used to perform pairwise reconstruction for each non-ambiguous (only one possible candidate) binocular stereo match [17]. The 3D reconstructed point is reprojected onto the remaining cameras in order to determine the set of corresponding 2D marker points in all camera views. A point is considered correctly reconstructed if it reprojects into at least one marker point of other camera view.

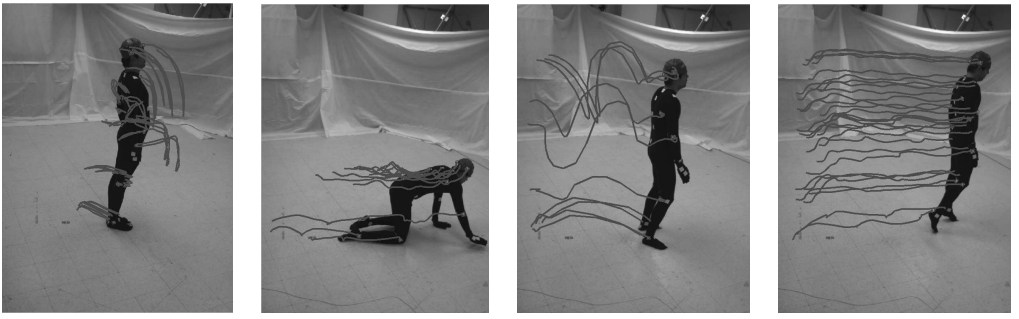
In our spatial correspondence algorithm, for each frame, a cloud of 3D points corresponding to marker locations is computed using reconstruction. A 3D point reconstruction is performed with stereo configurations of three cameras. In a circular arrangement of 8 cameras surrounding the scene, the stereo camera triplets used are (c_i, $c_{(i+1)\%8}$, $c_{(i+2)\%8}$), where $i = 1, \ldots, 8$. The set of 2D detected points in cameras c_i, $c_{(i+1)\%8}$, and $c_{(i+2)\%8}$ are denoted as D_i, $D_{(i+1)\%8}$, and $D_{(i+2)\%8}$, respectively. For each stereo triplet, every combination of three detected points (p_m, p_n, p_o) gives rise to a 3D point reconstructed through triangulation, where $p_m \in D_i$, $p_n \in D_{(i+1)\%8}$, and $p_o \in D_{(i+2)\%8}$.

The points p_m, p_n, and p_o are matched when the reprojection error of the reconstructed point is lower than a certain threshold. In this case, the associated 3D point is used as a cloud point which possibly represents a marker in 3D space. A confidence value for a cloud point is determined as the product of the reprojection error and the absolute intensity difference [18] for the corresponding reprojection points in each camera. The same process is performed with binocular stereo configurations to obtain cloud points from stereo pairs.

If the same detected 2D point is used to reconstruct two cloud points, only the one with higher confidence value is used. Additionally, points reconstructed from a trinocular configuration have a higher priority than from a binocular setup. If two cloud points are almost coincident, they could be merged or the one with lower confidence valued is filtered out. Cloud points reconstructed from a binocular configuration are more susceptible to errors and further filtering is performed according to the confidence value.

7 Temporal Correspondence

A rigid motion is a transformation consisting of translation and rotation which leaves a point arrangement unchanged. Our version of the temporal correspondence problem (tracking) concerns in establishing matches between two sets of sparse reconstructed 3D points in subsequent frames related by a non-rigid motion. A temporal concatenation of the markers position is performed frame by frame in order to reconstruct the 3D trajectory of most markers (see Fig. 10). In practice, the trajectories are not complete for all points through the full sequence. The problem involves registering two different configurations of the same articulated object which are represented by point clouds.



(a) Bow. (b) Crawl. (c) Jump. (d) Tiptoe.

Fig. 10: Marker trajectory for human activities.

The point clouds contain marker points and spurious points not representing any marker. In a point cloud, the marker points may not cover all markers in the marker set. The non-rigid point matching problem involves finding correspondences between the two point sets, discarding spurious points, and possibly computing a non-rigid mapping that can transform one point set onto the other.

Let U and V be two sets of 3D points representing markers in consecutive frames such that $U = \{U_1, \ldots, U_m\}$ and $V = \{V_1, \ldots, V_n\}$. In our temporal correspondence algorithm, the matching of each pair (U, V) of clouds requires the computation of a strength matrix $S_{m \times n}$, where each element $S(i, j)$ represents the support for a match between points (U_i, V_j). The strength function is composed by a reprojection distance and a similarity measure.

Let u_i^k and v_j^k be the reprojection points of U_i and V_j into camera k, the reprojection distance $d(U_i, V_j)$ is defined as the summation of the Euclidean distance between u_i^k and v_j^k for all cameras. The similarity measure $f(U_i, V_j)$ used is the summation of absolute intensity difference $\mathrm{IAD}(u_i^k, v_j^k)$ for all cameras. The strength function integrates different observations that may not be commensurate with each other. This way, the reprojection distance and similarity measure are normalized such that $S(i, j) = d(U_i, V_j)^w \times f(U_i, V_j)$, where w is a normalizing weight.

The winner-takes-all strategy [24] matches the points that have the maximum strength for both row and column of the strength matrix. If $S(i, j)$ is both the greatest element in its row and column, then points U_i and V_j are in one-to-one correspondence with one another. For each frame, the winner-takes-all algorithm computes the mapping $j = M(i)$ for the sets of points U and V. The trajectories of detected markers are computed from the point cloud mappings. The recorded Cartesian coordinates of points in 3D space are often called translational data. A trajectory is broken when the marker is occluded or the algorithm confuses one marker with another. This way, a trajectory is possibly incomplete and missing markers may exist.

8 Post-Processing

Most optical motion capture systems require human intervention for identifying markers through labeling, finding missing markers due to occlusion, and correcting possible errors detected in a rigidity test. The use of a kinematic human body model allows the association of markers with detected points and the inference of the position of missing markers. However, ambiguity issues persist and manual tasks are performed when the automatic techniques fail.

8.1 Labeling

Labeling (association problem) consists in determining which detected points correspond to each of the markers in the marker configuration [23]. All the markers need to be identified in each sequence frame in order to associate the skeleton model to the cloud of markers. The user has to associate each body marker with a 3D trajectory through a manual classification task. This way, the labeling process is performed only once when the marker trajectories are available. After an initial labeling in the starting frame, another approach executes labeling automatically at each frame and tracking is achieved as a byproduct. Fully automatic labeling may be achieved if the performing subject adopts a specific pose at the beginning of the sequence.

8.2 Missing Markers

An ideal 3D trajectory for a marker lasts for the whole sequence from beginning to the end. In general, trajectories are broken by occlusion which causes a point to be missing and lost in the tracking process. Occlusion is a significant problem in marker-based motion capture systems. Feature markers may be occluded by moving subparts of the actor or other objects in the scene. Self-occlusion occurs when rotation of a subpart makes a marker in this subpart invisible from the camera origin. Markers may often disappear by moving out of the camera field of view. Absence of a marker also happens when the feature is not measured due to sensory error. If enough cameras do not see a marker, triangulation becomes impossible and the 3D position cannot be calculated. Usually, increasing the number of

cameras mainly reduces occlusion effects while also improves accuracy and coverage. Accuracy does not improve much after two cameras cover the space [9].

A monocular reconstruction could be performed for the missing markers using the set of 2D points that are not associated to any reconstructed point [17]. For each unassociated 2D point in all cameras, the marker should be located on the ray r emanating from the camera center and passing through the 2D position. The distance Δ from the marker to another connected marker is given by the kinematic body model as the length of a rigid link. A similar distance may be computed from the previous or next frame. A sphere centered at the connected marker with radius Δ is intersected with the ray r at the reconstructed marker point. More connected markers and spheres may be used for disambiguation when the intersection leads to two points. The position of the marker in the previous or next frame could also be used to resolve the reflective ambiguity.

Articulated objects reveal their kinematic constraints between two connected subparts. Kinematic constraints enforce topology, link length constancy based on a rigidity assumption, and restricts joint motion to rotation about a fixed axes in revolute joints. A 3D kinematic model restricts the 3D pose to a number of isolated candidate regions in the joint state space corresponding to discrete combinations of reflective ambiguities at each link [12] (see Fig. 11).

Fig. 11: Reflective ambiguity and joint angle limits.

Joint angle range specify the limits to which revolute joints can rotate about their corresponding axes for a kinematic model. Only physically valid solutions for 3D motion satisfy joint angle limitation constraints. Limits on the joint angle range are represented by inequalities such as $q_k \geq \theta_k$, where q_k is the k^{th} angle parameter and θ_k is the lower limit for q_k.

Another strategy to find the missing markers uses the location of two (three) previous frames to compute the point velocity (acceleration). The point velocity (acceleration) is applied to estimate the position of the marker in the current frame.

8.3 Rigidity Test

The motion of some markers is difficult to collect when the corresponding body feature slides under the skim. For this reason, post-processing is required to correct possible

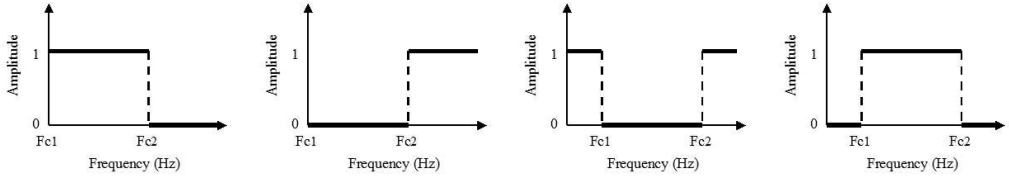
errors before converting translational data into hierarchical rotational information. These errors are detected through a rigidity test and manually corrected.

Given a human body model as a structure with rigid connected segments, the distance between two markers on the same segment is contained within an interval $[l-\varepsilon, l+\varepsilon]$, where l is the mean length of the rigid segment. For all markers on the same segment, a segment rigidity test [17] is performed to check if the distance d between these markers is in the interval: $|d-l| < \varepsilon$. If the distance d is larger than the distance specified by the model, one marker is moved to an acceptable distance from the other marker. This process is reliable if the variance of the measured distances for segment length is not large.

8.4 Motion Data Filtering

Filtering is the reduction or amplification of certain components of a signal in either the time or the frequency domain [26]. In the frequency domain, filtering changes the Fourier coefficients of some frequency components according to a transfer function. Using cutoff frequencies Fc_1 and Fc_2, the transfer function of the filter has three frequency spectrum bands: pass $[0, Fc_1]$, transition $[Fc_1, Fc_2]$, and stop $[Fc_2, \infty]$. The pass band allows all the covered frequency components to remain unchanged. The transition band decreases the power of the frequency components. The stop band reduces or eliminates all the remaining frequencies. The filtering process involves the multiplication of the power spectral density by the transfer function. The selection of an appropriate cutoff frequency is an important problem, since a high cutoff allows noise to pass while a low cutoff could eliminate part of the signal.

A square filter has no transition band since the transfer function is 1 for certain frequencies and 0 for the remaining frequencies. According to the eliminated frequencies, different types of square filters include low pass filter to eliminate higher frequencies, high pass filter to eliminate lower frequencies, band stop filter to remove the central components, and band pass filter to keep only the central part of the power spectrum (see Fig. 12). In human movement data, the signal is predominant in lower frequencies and noise is in higher frequencies, consequently, motion filtering uses low pass filter.



(a) Low pass. (b) High pass. (c) Band stop. (d) Band pass.

Fig. 12: Types of square filters.

The low pass Butterworth filter [33] is applied to a forward time series $\langle x_1, x_2, \ldots, x_n \rangle$ obtaining a filtered time sequence $\langle y_1, y_2, \ldots, y_n \rangle$. The filter is then reapplied to the reverse

filtered time series $\langle y_n, y_{n-1}, \ldots, y_1 \rangle$. The order of the Butterworth filter indicates the sharpness of the transfer function. In motion data, a second order filter works reasonably well. This filtering requires extrapolation techniques to create data points near the edges of the signal since each filtered data point requires previous and future data points.

9 Rotational Data

The local coordinate system of a rigid segment in space is completely defined by three points in the segment. This way, given at least three markers on a segment, the position (translation) and orientation (rotation) of that segment in space can be found with respect to the world reference coordinate system by computing the alignment transformation between local and world coordinate systems.

Each subpart in the kinematic model is characterized by a set of attached feature markers whose location fully describes the position and orientation of the subpart. The grouping of specific markers in each segment allows the computation of internal joint rotations. Our human body model has 21 rigid subparts forming a hierarchical structure (see Table 3). The skeleton model has 60 degrees-of-freedom (30 joints) and six parameters (position and orientation) in 3D space.

Table 3. Rigid subparts and defining markers.

Subpart	Marker I	Marker II	Marker III
Pelvis	R_HIP+L_HIP	SACRUM	L_HIP
Lumbar Spine	L1	SACRUM	R_HIP+L_HIP
Lower Thoracic	T7	L1	STERNUM
Middle Thoracic	T4	T7	STERNUM
Upper Thoracic	T1	T4	STERNUM
Cervical Spine	NUCHALE	T1	STERNUM
Head	FOREHEAD	NUCHALE	L_HEAD
Right Upper Leg	R_HIP	RE_KNEE	RI_KNEE
Right Lower Leg	RE_KNEE+RB_KNEE	RE_ANKLE+RI_ANKLE	RI_KNEE+RF_KNEE
Right Foot	RE_ANKLE+RB_ANKLE	RI_ANKLE+RF_ANKLE	R_TOE
Left Upper Leg	L_HIP	LE_KNEE	LI_KNEE
Left Lower Leg	LB_KNEE	LE_ANKLE+LI_ANKLE	LI_KNEE+LF_KNEE
Left Foot	LE_ANKLE+LB_ANKLE	LI_ANKLE+LF_ANKLE	L_TOE
Right Clavicle	STERNUM	R_SHOULDER	R_AXILLA
Right Upper Arm	R_SHOULDER	RE_ELBOW	RI_ELBOW
Right Lower Arm	RE_ELBOW	RU_WRIST	RF_ELBOW+RI_ELBOW
Right Hand	RU_WRIST	R_HAND	RR_WRIST
Left Clavicle	STERNUM	L_SHOULDER	L_AXILLA
Left Upper Arm	L_SHOULDER	LE_ELBOW	LI_ELBOW
Left Lower Arm	LE_ELBOW	LU_WRIST	LF_ELBOW+LI_ELBOW
Left Hand	LU_WRIST	L_HAND	LR_WRIST

A kinematic model is organized in nested coordinate systems forming a hierarchical tree (see Fig. 13). A marker close to the center of gravity is used as the root of a body-based hierarchical chain of nodes, where each node represents a subpart in the model. The root

marker is associated with the global translational data and orientation for the body. The spatial state for all subparts other than the root are relative to the parent node local coordinate system. The rotation and translation of a subpart are computed as the transformation which aligns the local coordinate system with the parent coordinate system in the same way as discussed in Section 4.2. The angles corresponding to the transformation between coordinate systems of connected subparts are called rotational data.

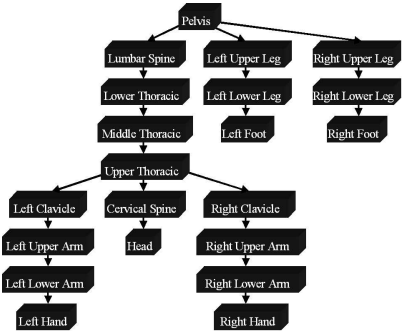


Fig. 13: A hierarchical organization for the kinematic model.

Representations for the rotational part of a homogeneous transformation include axis-angle, quaternion, rotation matrix, and Euler angle. Usually, the Euler angle form is more intuitive and rotation matrices are converted to this representation. The Euler angle representation depends on the choice of a rotation axis sequence which leads to a definition convention. For the rotation axis sequence *Z-Y-Z* and the respective angles ϕ, θ, ψ, the rotation matrix is $R(\phi, \theta, \psi) = R_Z(\phi) \times R_Y(\theta) \times R_Z(\psi) =$

$$\begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -\sin\phi\sin\psi + \cos\phi\cos\theta\cos\psi & -\sin\phi\cos\psi - \cos\phi\cos\theta\sin\psi & \cos\phi\sin\theta \\ \cos\phi\sin\psi + \sin\phi\cos\theta\cos\psi & \cos\phi\cos\psi - \sin\phi\cos\theta\sin\psi & \sin\phi\sin\theta \\ -\sin\theta\cos\psi & \sin\theta\sin\psi & \cos\theta \end{bmatrix}.$$

In the *Z-Y-Z* convention, the inverse conversion to Euler angles from a rotation matrix is found as $\phi = \text{atan2}(R_{2,3}, R_{1,3})$, $\theta = \text{atan2}(\cos\phi R_{1,3} + \sin\phi R_{2,3}, R_{3,3})$, and $\psi = \text{atan2}(-\sin\phi R_{1,1} + \cos\phi R_{2,1}, -\sin\phi R_{1,2} + \cos\phi R_{2,2})$.

The gimbal lock problem occurs when Euler angles are used to represent rotations around the axes. The loss of one degree of rotational freedom results in the fact that a rotation does not occur due to the alignment of the axes.

The joint local position and orientation are defined by a transformation matrix from a global referential to the local one. This matrix is computed recursively by multiplying all the transformation matrices that correspond to the preceding joints in the body hierarchy tree until the root of the hierarchy is reached. For example, the forearm depends on the upper arm which depends on the shoulder and so on.

Unlike translational data, which may improve with smoothing, rotational data should NOT be smoothed, since that would eliminate the variational detail that gives the motion its realism and accuracy [7].

10 Data File Formats

Whether the motion capture system is proprietary or commercially available, the motion data has to be useful to the widest number of potential users. Porting data sets is an important issue and involves the use of file formats for motion data. The rotation evaluation ordering is one of the subtlest problems in porting data sets. Especially for the root node of the skeleton, the motion may appear to be correct even when interpreted with the wrong rotation order.

Besides rotation ordering, there are issues associated with orientation of the axes relative to the bone: bone direction (X, Y, Z, or arbitrary), global (axes rotate with the bone) or local (axes remain fixed relative to its orientation) rotations, rotation inheritance from parent or relative to the origin. These issues need to be considered when porting data and must be handled prior to inputting the data.

The Biovision BVA/BVH formats and the Acclaim Motion ASF/AMC formats are data file formats for motion capture data [19]. These file formats are usually supported by motion capture systems, animation packages, and other software in order to store, export, and import motion capture data in different programs. In this section, we present a brief description of these file formats.

In a BVA file there is a segment block for each rigid segment which consists of a segment name, number of frames, and a frame time (see Fig. 14). There are 9 channels that describe the position (translation), orientation (rotation), and scale (length of the segment). A description of the channels and respective measure units are included as a header. For each frame, the actual motion data corresponds to a line with values for each channel.

```
Segment:    Pelvis
Frames:     2
Frame Time: 0.083333
```

XTRAN	YTRAN	ZTRAN	XROT	YROT	ZROT	XSCALE	YSCALE	ZSCALE
INCHES	INCHES	INCHES	DEGREES	DEGREES	DEGREES	INCHES	INCHES	INCHES
0.38	12.35	36.88	-78.14	46.29	-2.16	4.16	4.16	4.16
8.17	10.35	47.86	-94.12	54.97	-4.44	4.16	4.16	4.16

Fig. 14: A sample BVA file.

A BVH file stores motion for a hierarchical skeleton such that the motion of a child segment depends on the parent segment (see Fig. 15). The word HIERARCHY is followed by a skeleton definition section. The first segment defined is the ROOT and each segment in the hierarchy is defined as a JOINT. All joints within a pair of braces are the children of a parent joint. Each braced block has an OFFSET describing displacement from its parent and a CHANNELS definition for the parameters (number of channels and data types for each

channel) of that segment. The leaves of the hierarchy are `End Site` joints whose offset determines the length of the segment. After the skeleton definition section, the MOTION section describes the number of frames, the time for each frame, and the movement over time for all segments and channels at once.

```
HIERARCHY
ROOT Pelvis
{
    OFFSET          0.00      0.00      0.00
    CHANNELS 6 Xposition Yposition Zposition Zrotation Xrotation Yrotation
    JOINT RightUpperLeg
    {
          OFFSET     0.00      5.21      0.00
          CHANNELS 3 Zrotation Xrotation Yrotation
          JOINT RightLowerLeg
          {
              OFFSET           0.00      5.45      0.00
              CHANNELS 3 Zrotation Xrotation Yrotation
              End Site
              {
                    OFFSET     0.00      3.87      0.00
                }
          }
    }
}
MOTION
Frames:     2
Frame Time: 0.083333
8.03 35.01 88.36 -3.41 14.78 -164.35 13.09 40.30 -24.60 7.88 43.80 0.00
7.81 35.10 86.47 -3.78 12.94 -166.97 12.64 42.57 -22.34 7.67 43.61 0.00
```

Fig. 15: A sample BVH file.

An Acclaim motion capture data has one file describing the skeleton hierarchy (ASF) and another file with the motion data (AMC). An ASF file has many sections, each one starting with a keyword preceded by a colon (see Fig. 16). The `:units` section describes all values and units of measure used. The `:root` section contains the `axis` and `order` elements specifying the order of operations for the initial offset and transformation. The `position` (`orientation`) element describes the root translation (rotation). The `:bonedata` section describes the remaining bones delimited by `begin` and `end` statements. The `direction` vector describes the initial position. The `axis` vector represents global orientation and `XYZ` specifies the order of rotations. The `dof` parameter describes the possible degrees of freedom: translation (`tx`, `ty`, `tz`), rotation (`rx`, `ry`, `rz`), and stretch in length (`l`). Each degree of freedom corresponds to a channel in the AMC file. The `limits` element describes the limits of the degrees of freedom. Finally, the `:hierarchy` section specifies the hierarchy of the bones. Each line represents a parent bone followed by its children.

```
:version 1.10
:name BioSkeleton
:units
   mass 1.0
   length 1.0
   angle deg
:documentation
   Example of an ASF file
:root
   axis XYZ
   order TX TY TZ RZ RY RX
   position 0.0 0.0 0.0
   orientation 0.0 0.0 0.0
:bonedata
begin
   id 1
   name rightupperleg
   direction 0.000000 1.000000 0.000000
   length 1.000000
   axis 0.00000 0.00000 0.00000 XYZ
   dof rx ry rz
   limits (-180.0 180.0)
      (-180.0 180.0)
      (-180.0 180.0)
end
:hierarchy
begin
   root rightupperleg
   rightupperleg rightlowerleg
end
```

Fig. 16: A sample ASF file.

The AMC file contains the motion data for each frame (see Fig. 17). The frame number is declared and followed by each bone name and data for each channel defined for that bone.

```
:fully-specified
:degrees
1
root 12.0 33.0 45.0 0.0 90.0 45.0
rightupperleg 0.0 0.0 0.0
2
root 12.0 33.0 45.0 0.0 90.0 45.0
rightupperleg 0.0 0.0 0.0
```

Fig. 17: A sample AMC file.

11 Visual Hull Reconstruction

Silhouette images from multiple viewpoints obtained from background subtraction are useful in reconstructing 3D geometry from 2D images. Two approaches for visual hull

reconstruction from silhouettes are volumetric and polyhedral. Volumetric reconstruction tessellates the scene space volume by subdividing the space into a regular grid of volume elements (voxels). Each voxel is projected into every image plane. Projections can be pre-computed and stored in a look-up table in order to improve computation performance. The voxel whose projection falls outside of any silhouette is carved away. A voxel is classified as occupied space if it projects into all foreground silhouettes (see Fig. 18).

Fig. 18: Volumetric reconstruction for a kneel action.

Polyhedral reconstruction extracts contours from silhouette images. Given camera calibration, each silhouette contour is projected into 3D space as a generalized cone (polyhedral representation) containing the actual 3D object. The 3D intersection of these cones from all viewpoints gives a polyhedral visual hull.

12 Inverse Kinematics and Dynamics

Inverse kinematics allows the control of a 3D character's limbs by treating them as a mechanical linkage (kinematic chain). Control points, connected to the ends of these chains, allow the entire chain to be manipulated by a single "handle". In the case of an arm, the handle would be the hand at the end of the arm.

These control points can also be driven by external data. In other words, inverse kinematics allows the artist to design a skeleton structure that can be driven from data sets created by a motion capture system. Even capture systems with a limited number of sensor points can animate a more complex structure through the use of inverse kinematics.

The problem is usually solved by loosely connecting the data set to the control points. We then allow the incomplete data set to influence the control points, but impose constraints and limits on the motion of the skeletal structure that the inverse kinematics solution must obey. As we increase the number of points captured, we depend less on inverse kinematics and more on the loose coupling mechanism. If we were able to track enough sensors, we would have no need for inverse kinematics.

A human is an actuated system that uses muscles to convert stored energy into time-varying forces and torques acting on joints. The inverse dynamics problem concerns in finding the force and torque that generate the motion at each joint. Newton's second law $F = ma$ describes the linear motion of the rigid body, where F is the vector force on the body with mass m and linear acceleration a. The Euler's equation $N = J\alpha + \omega \times J\omega$ governs the rotation of the rigid body, where N is the vector torque on the body with inertia matrix J about its center of mass, α is the angular acceleration, and ω is the angular velocity.

13 Manipulation and Editing

Stylizing, exaggerating, and altering motion data for character animation is part of the potential for using motion capture. An example would be to apply a rotation or translation to the captured motion, so that the character takes a different path through the environment. Inverse kinematics techniques can be used to affect the motion. For instance, lifting the hands of a runner while still maintaining the realism of the captured motion. Crowd scenes can be created by duplicating motion many times for crowd members, with slight variations.

The use of motion capture data implies the need for a suite of tools able to dissect and manipulate motion data sets in a variety of ways, to cut and paste motions together seamlessly, and to interpolate one motion into another without losing the essential nature of each motion. Editing operations include loop, blend, offset, lengthen, shorten, dampen, add to, or subtract from the motion.

Once the real motion is captured, the measured movement is converted into a virtual character motion. The application of motion from a real actor to an animated character, known as motion retargeting, has potential problems of scale, since the performers may be sized or built differently. This can become extreme when the relative proportions of the two characters are vastly different. The animated character may differ considerably from the performing actor in shape and proportion causing movement distortion [15]. The more different the character is in scale and proportion from the real performer, the more artifacts (e.g. self-collisions) the conversion introduces.

The motion must be automatically adjusted to maintain foot and hand location to avoid common problems like foot sliding and hands passing through objects. If a real performer has shorter arms than a virtual character, when the performer puts his/her hands together, the character's hands would pass through each other. One strategy to solve this problem is to keep the hands exactly where they were when captured, and adjust the angle on the elbows to compensate. If the character's arms are too short to reach an object, then the reach is impossible since Physics laws cannot be changed.

A more general motion retargeting problem involves the transferring of the motion of a human actor to an inanimate object. Using a chair as a target example, the problem consists in mapping the motion from a human character (two legs) to an object of four legs.

14 Conclusions

This tutorial is a result of research on optical motion capture towards the creation of a human activity database with visuo-motor information at the Computer Vision Laboratory of the University of Maryland. We designed and implemented a new **M**o**C**ap system, named after a northeastern Brazilian cactus tree: **M**anda**C**aru [5].

In our motion capture system, we used an architecture which divides the problem into four sequential modules: marker detection, spatial correspondence (stereo matching), temporal correspondence (tracking), and post-processing. The post-processing is further divided into four sub-modules: labeling, missing markers location, rigidity enforcement, and filtering. Each sub-problem of our architecture and the corresponding novel techniques used to solve them were described in this tutorial.

Based on this architecture, we suggest a Matlab® toolbox for Optical Motion Capture where each module version may be implemented in order to consider different constraints. For example, the marker detection module may consider color images instead or grayscale images. Other methods may also be used in the solution of sub-problems. The application of snakes to the computation of silhouette perimeter could be considered towards more robust marker detection. Volumetric reconstruction is another technique to be explored in the future.

References

[1] Humanoid animation working group.

http://www.h-anim.org/

[2] CAESAR: Civilian American and European surface anthropometry resource project.

http://www.sae.org/technicalcommittees/caesarhome.htm

[3] Multi-camera self-calibration.

http://cmp.felk.cvut.cz/~svoboda/SelfCal/

[4] Camera calibration toolbox for Matlab.

http://www.vision.caltech.edu/bouguetj/calib_doc/

[5] MandaCaru: An optical motion capture system.

http://www.cs.umd.edu/~guerra/OptMoCap.html/

[6] Arun, K., Huang, T., and Blostein, S. 1987. Least-squares fitting of two 3-D point sets. *IEEE Transactions on Pattern Recognition and Machine Intelligence* 9(5): 698-700.

[7] Bodenheimer, B., Rose, C., Rosenthal, S., and Pella, J. 1997. The process of motion capture: Dealing with the data. In Proc. of the Eurographics Workshop on Computer Animation and Simulation, Wein, Austria, 3-18.

[8] Borghese, N., Di Rienzo, M., Ferrigno, G., and Pedotti, A. 1990. Elite: a goal-oriented vision system for moving objects detection. *Robotica* 9: 275-282.

[9] Chen, X., and Davis, J. 2000. Camera placement considering occlusion for robust motion capture. Technical Report CS-TR-2000-07, Stanford University.

[10] 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 Computer Vision and Pattern Recognition, Hilton Head Island, SC, vol. 2: 714-720.

[11] Cutler, R., Duraiswami, R., Qian, J., and Davis, L. 2001. Design and implementation of the University of Maryland Keck Laboratory for the analysis of visual movement. Technical Report CS-TR-4329, University of Maryland.

[12] DiFranco, D., Cham, T., and Rehg, J. 2001. Reconstruction of 3-D figure motion from 2-D correspondences. In Proc. of Computer Vision and Pattern Recognition, Kauai, HI, vol. 1: 307-314.

[13] Faugeras, O. and Robert, L. 1996. What can two images tell us about a third one? *International Journal of Computer Vision* 18: 5-19.

[14] Fischler, M. and Bolles, R. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. 1981. *Communications of the ACM* 24(6): 381-395.

[15] Gleicher. M. 1998. Retargeting motion to new characters. In Proc. of Computer Graphics and Interactive Techniques, Orlando, FL, 33-42.

[16] Hartley, R. and Zisserman, A. 2000. Multiple view geometry in computer vision. Cambridge University Press, Cambridge, UK.

[17] Herda, L., Fua, P., Plankers, R., Boulic, R. and Thalmann, D. 2001. Using skeleton-based tracking to increase the reliability of optical motion capture. *Human Movement Science Journal* 20: 313-341.

[18] Kanade, T. 1994. Development of a video-rate stereo machine. In Proc. of ARPA Image Understanding Workshop. Monterey, CA, 549-558.

[19] Lander, J. 1998. Working with motion capture file formats. *Game Developer* 5(1): 30-37.

[20] Menache, A. 1995. Understanding Motion for Computer Animation and Video Games. Morgan Kaufmann, San Francisco, CA.

[21] Morris, D. and Reehg, J. 1998. Singularity analysis for articulated object tracking. In Proc. of Computer Vision and Pattern Recognition, Santa Barbara, CA, 289-296.

[22] Ogale, A. 2004. The compositional character of visual correspondence. Ph.D. Thesis. Computer Science Department, University of Maryland.

[23] Ringer, M. and Lasenby, J. 2000. Modelling and tracking articulated motion from multiple camera views. In Proc. of the British Machine Vision Conference, Bristol, UK, 172-182.

[24] Rosenfeld, A., Hummel, R., and Zucker, S. 1976. Scene labeling by relaxation operations. *IEEE Transactions on Systems, Man, and Cybernetics* 6(6): 420-433.

[25] Shimada, N., Shirai, Y., Kuno, Y., and Miura, J. 1998. Hand gesture estimation and model refinement using monocular camera – ambiguity limitation by inequality constraints. In Proc. of Automatic Face and Gesture Recognition, Nara, Japan, 268-273.

[26] Stergiou, N. 2004. Innovative analyses of human movement. Human Kinetics, Champaign, IL.

[27] Sturm, P. and Triggs, B. 1996. A factorization based algorithm for multi-image projective structure and motion. In Proc. of the European Conference on Computer Vision, Cambridge, UK, 709-720.

[28] Svoboda, T. 2003. Quick guide to multi-camera self-calibration. Technical Report BiWi-TR-263, Computer Vision Lab, Swiss Federal Institute of Technology.

[29] Theobalt, C., Li, M., Magnor, M., and Seidel, H.-P. 2003. A flexible and versatile studio for synchronized multi-view video recording. In Proc. of Vision, Video and Graphics, Univeristy of Bath, UK, 9-16.

[30] Tsai, R. 1986. An efficient and accurate camera calibration technique for 3d machine vision. In Proc. of Computer Vision and Pattern Recognition, Miami, FL, 364:374.

[31] Tsai, R. 1987. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf cameras and lenses. *IEEE Journal of Robotics and Automation*, RA-3(4): 323-344.

[32] Weik, S. and Liedtke, C.-E. 2001. Hierarchical 3d pose estimation for articulated human body models from a sequence of volume data. *Lecture Notes in Computer Science* 1998: 27-34.

[33] Winter, D., Sidwall, H., and Hobson, D. 1974. Measurement and reduction of noise in kinematics of locomotion. *Jornal of Biomechanics* 7: 157:159.