Converting Sequences of Human Volumes into Kinematic Motion

Chi-Wei Chu, Odest Chadwicke Jenkins, Maja J Matarić
Robotics Research Laboratories
Department of Computer Science
University of Southern California
Los Angeles, CA 90089-0781
chuc,cjenkins,mataric@usc.edu

Abstract

We present an approach for producing articulated motion for human from a sequence of 3D points forming the volume of a human. In our approach, we use an existing voxel carving technique for collecting point volumes consisting of a single individual. We propose a volume-to-posture procedure based on the estimation of a skeleton curve. A skeleton curve is a underlying "wire" structure for the volume is approximated using a technique we developed, called nonlinear spherical shells. Nonlinear spherical shells uses Isomap, a nonlinear dimension reduction function, with clustering, and interpolation to find points for the skeleton curve of the captured volume. Using the volume and the skeleton curve, we find rigid bodies for the volume that conform to a given hierarchy and convert the configuration of these bodies into a kinematic posture.

1 Introduction

The ability to collect information about people habitating a certain environment is a familiar problem in computer vision and can be useful for a variety of applications. Typical applications of human activity sensing include surveillance of an area for various types of behavior, perceptual user interfaces for intelligent or interactive spaces, and collection of data for later analysis or processing. However, the type of data provided from sensing techniques may not be readily amenible for use in such an application. Our aim in this work is to provide a bridge between reasonable representations for sensed human data and human data useful for further applications and analysis.

In order to address this problem, we must first state our expectations for both the sensing and application data representations. Towards the end of activity recognition, we expect the sensed data to provide a representative description of an unmarked and uninstrumented human. More to the point, the sensed data representation should not rely on obtrusive sensing mechanisms, even if the humans are actually instrumented. For applications, we expect data representation to provide a familiar and relatively intuitive specification of human motion.

Fortunately, suitable representations for both sensed and application human data are readily available. Through passive sensing (e.g., cameras) and techniques such as voxel carving [9], a sampling of 3D points comprising the volume of a human can be readily sensed, even at interactive rates. This representation is also desireable because various types of passive sensing technologies have the potential for producing data in this representation. For the application data, we will represent human motion using a kinematic model, assuming only the topology of the rigid links. Kinematics is standard representation of articulated motion in many applications, including humanoid robotics and character animation.

In this paper, we present an approach, outlined in figure 1, for deriving kinematic motion and rigid link parameters from a sequence of 3D point volumes of a human. The core of our approach is a volume-to-posture procedure that is applied to each volume in the squence. The volume-to-posture procedure takes as input a set of 3D points representative of the sensed human and procduces an estimation of the length, width, and joint angle parameters of each rigid link in a known kinematic topology.

Our approach is similar to work by Mikic et. al. [8] in that both methods are converting a volume of points into human kinematics. However, their approach is based on statistical tracking of a body model initialized by fitting a known human body template to the first frame of motion. In contrast, our approach focuses on the geometric characteristics of the volume by introducing an intermediate step of finding *principal curves*, called *nonlinear spherical shells*. We treat a set of principal curves, formed into a *skeleton curve*,

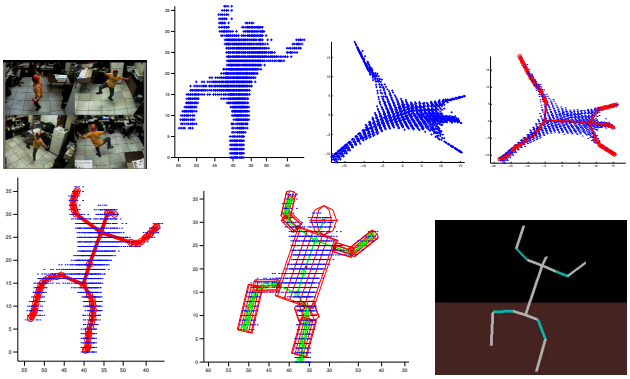

Figure 1: The outline of our appraoch. A human viewed in multiple cameras (1) is used to build a point volume (2). This volume is transformed into a topology accentuating embedding (3). A skeleton curve (4) are found in this embedding and interpolated back to the input space (5). The skeleton curve allow for the fitting of body parts (6). The kinematic posture is generated from these world-space body parts (7).

as the underlying structure of the volume, similar to a wire frame support structure embedded in a soft model or puppet. This skeleton curve serves as a guide for dividing the points of the volume into body parts. We are then able to construct postures from these body parts. By constructing the skeleton curve, we *(i)* can derive accurate kinematic postures for each frame in a motion sequence due to the relative speed of nonlinear spherical shells *(ii)* have the potential to extend our approach to derive the kinematic structure of various creatures with unknown kinematic topologies.

We have implemented a working system of our approach. We demonstrate the usefullness of our implementation by applying point volume sequences of different human motion. The derived kinematic motion was exported to a standard motion capture format and used to actuate the upper-body of a 20 DOF dynamic humanoid simulation.

2 Volume Capture

In this section, we describe our implementation of an existing volume capture technique using multiple calibrated cameras. While not the focus of our work, this implementation does provide an adequate means for collecting input volume data. This implementation is derived from the work of Penny et. al. [9] for real-time volume capture; however, several other approaches are readily available. In our capture setup, we place multiple cameras around three sides of a hypothetical rectangular volume, such that each camera can view roughly all of the volume. This rectangular volume is a voxel grid that divides the space in which moving objects can be captured.

The intrinsic and extrinsic calibration parameters for the cameras are extracted using Camera Calibration Toolbox designed by [2]. The intrinsic parameters such as skew and distortions coefficients enable us to compensate the image distortions. The extrinsic parameters provide a common world coordinate system to relate the viewpoint of each camera to the voxel grid. With these parameters, given an arbitrary point coordinate in 3D space, we can easily calculate the pixel coordinate on the four image planes that the point projects to. Given this capability, we precompute a lookup table relating each voxel to its corresponding pixel location in each camera image.

For each set of synchronized camera images, silhouettes of active objects in the voxel grid are segmented in each image and used to carve the voxel grid. The segmentation of camera images into silhouettes is performed by our implementation of an algorithm proposed in [4]. Each voxel in the grid can then be determined to be part of an active object or not by counting and thresholding the number of camera images in which it is part of a silhouette. One set of volume data is collected for each set of synchronized images and is stored for offline processing.

Our implementation of the image processing and volume capturing algorithm is not perfect, but is also not the focus of our work. Consequently, some silhouette images or volume data may need manual modification before used in the following motion capure algorithm. We discuss some of the shortcomings in this implementation within Section 5.

3 Skeleton Curve Determination for Human Volumes

The first step of our volume-to-posture procedure is to estimate a set of points that reside on the *principal curves* of the volume. The definition of principal curves can be found in [5] or [6] as "self-consistent" smooth curves which pass through the "middle" of a d-dimensional data cloud, or nonlinear principal components. Our procedure ignores the necessary smoothness properties and focuses placing approximate principal curves through the middle of the volume, similar to a wire spine or support structure of a puppet. These curves will be of significant use in our ability classify voxels into body parts such as head, torso and limbs.

Even though the volume data in each frames will be different due to kinematic motion, the underly-

ing topological structure of the kinematics remains the same. We use this assumption for our principal curve approximation approach called *nonlinear spherical shells*.

With this concept in mind, we first apply *Isomap* [10], a nonlinear dimension reduction algorithm, to volume data of each frame. The purpose of using Isomap is to transform the volume data such that the effect of joint angles are removed and the underlying topology of the volume is preserved. In applying Isomap, we only need to specify the d dimensionality of the embedding and a function for determining a local neighborhood of points. An example result from an Isomap embedding of a volume is shown in Figure 2.

The embedded point volume will be roughly zero-centered in the transformed space, i.e., the origin of the embedding is located roughly at the waist of the volume. The "posture" of the embedded volume assumes a "da Vinci" like pose, with the arms and legs streched straight and outward from the embedding origin.

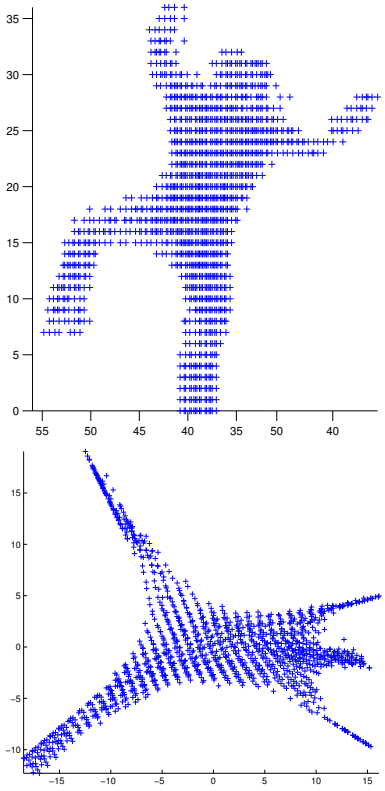

Figure 2: *Volume data in original 3D space (left) and Volume data in dimensionally reduced feature space. The dimension of the feature space is set to 3 (right)*

We can then leverage volumes embedded in this manner by splitting the embedded points into *concentric spherical shells*. A series of *concentric spheres* are placed on the embedded volume data, centered at the zero mean. These concentric spheres divide the volume data into spherical shells as shown in (Fig.3). Because the embedded volume takes on the "da Vinci" pose, the concentric spheres will provide good slices of the limb extremeties and distinguishable mid-body features. The points within each spherical shell are isolated and bounding box clustering, as in [3], is used on each shell to divide the points. Each cluster represents a limb and the centroid of each cluster is computed. The cluster centroids are considered as points on a principal curve associated with the points of the cluster.

The appropriate number of spherical shells and their size is dependent on the resolution of voxel grid. More shells will not necessarily provide better principal curves. If the interval between shells is less than the voxel resolution, many shells will contain zero or very few voxels and erratic principal curves will be produced.

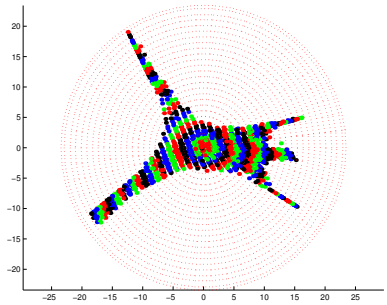

Figure 3: *Shells placed on the feature space data*

By linking principal curve points of overlapping clusters on adjacent spherical shells, a tree structure of principal curve points results, as shown as (Fig.4). We use this tree structure to help use classify voxel into rigid body parts in the next section.

The structure of the estimated principal curves may not immediately useful due to a variety of artifacts. We use a refinement procedure to prune and merge subtrees in the principal curves into a useful *skeleton curve*.

The refinement procedure begins by merging small branches that occur due to noise or smaller parts of the body (e.g., hands and feet). This operation should result in only 5 remaining leaves in the skeleton curve. Next, a new root node is formed by nodes for shells in the middle of the body with no articulation, i.e., a single encompassing cluster. Decendants of this root node that are also ancestors of limb branches are merged

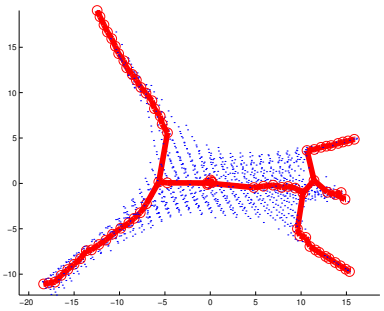

Figure 4: *The initial tree structure from shell clusters*

into a single branch node (e.g., chest and pelvis nodes). This following pseudocode describes in more detail the refinement procedure.

```
PROCEDURE Adjust_Tree_Nodes
BEGIN
  // Clip the noise branch and merge leaves of
  // palms and feet
  FOREACH node n IN tree t
    IF n.child > 2
      IF all of n.child have depth <= threshold
        merge all children of n into one node
      ELSE
        FOREACH c IN n.child AND
        c.depth <= threshold
          clip_subtree(c);
        END
      END
    END
  END

  // Merge root clusters until reach
  // first branch
  n = t.root;
  WHILE n.child == 1
    merge_node(n.child, n);
  END

  // Merge nodes in the chest or pelvis
  FOREACH leaf p IN tree t
    WHILE p.child == 1 OR p.parent != t.root
      p = p.parent;
    END
    // p is the root of the limb subtree after
    // the loop
    WHILE p.parent != t.root
      merge_node(p, p.parent);
    END
  END
END
```

Once points skeleton points are found in the embedded space, we then use radial-basis function weighted averages (local neighborhood) [1] to interpolate skeleton points to the original input space volume, maintaining the same tree-structure. Because each node in the principal curve represents a cluster of points, merged nodes are repositioned using the centroid of the points in the merged clusters. A sample skeleton curve is shown in (Fig.5).

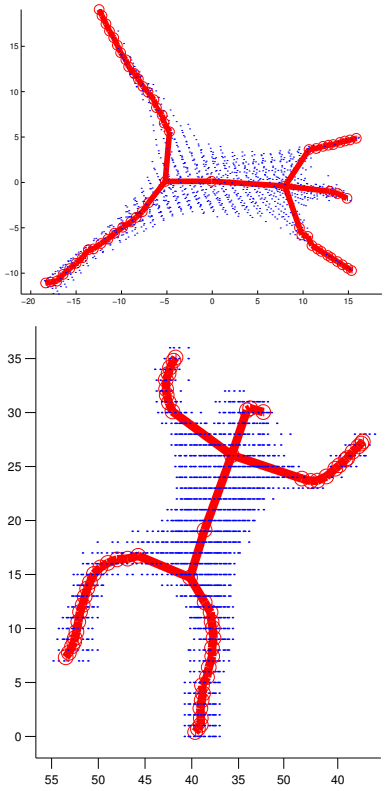

Figure 5: *The resulting tree structure in feature space (left) and in original data space (right)*

4 Kinematic Posture Estimation

The last part of the volume-to-posture procedure is the skeleton curve to find rigid body parts and their local transformations in the kinematic hierarchy. Using the five leaf skeleton curve as a guide, we can divide the volume points into rigid bodies and determine the kinematic posture of the human. The skeleton curve has five leaf branches. The shortest of these branches is classified as the head. The branches closes to the head branch are classified as arms, leaving two remaining

leg branches. Volume points are separated into one of these branches, the chest branch, the pelvis branch, or the root torso based on their cluster association to a skeleton curve point.

The head branch of the skeleton curve is used to fit a sphere to the points of the head branch. The center of the head sphere is determined by merging the skeleton curve points of the head branch. The radius of the head sphere is determined by the largest distance between its center and a head volume point. Volume points residing within this sphere are reassociated to the head, regardless of their previous classification. (Fig.6)

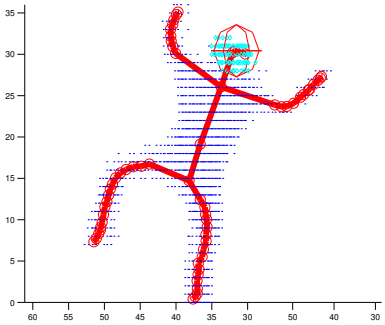

Figure 6: *Segmentation of the head volume points*

Next, a cylinder is grown to fit the root torso using a cylinder. The vector between the chest and pelvis nodes is the axis of the cylinder. To initialize the cylinder, the height and radius are set to bound the voxels of the root torso. The top of the cylinder is extended to meet the head sphere. The bottom of the cylinder is extended to bound the points of the pelvis. (Fig.7) Voxels that fall within the cylinder are reassociated to the torso regardless of their previous classification.

Unlike the head and torso, the arm and leg limb branches will require a slightly more sophisticated fitting procedure because they contain a two body articulation. This fitting procedure will place two cylinders on a limb for the upper and lower parts of the extremity. The procedure begins by ingoring the section of the skeleton curve residing within the torso cylinder. The last skeleton point of each limb branch within the torso cylinder is used as the placement of the limb root, such as the hip or shoulder joint.

The division of the volume points for a limb is determined by a separating skeleton point, having the maximum distance from the line segment connecting the limb root and the leaf of the limb branch. If this maximum distance falls below a threshold, the limb is extended straight, exhibiting no articulation. Limbs in this configuration cannot be divided into upper and

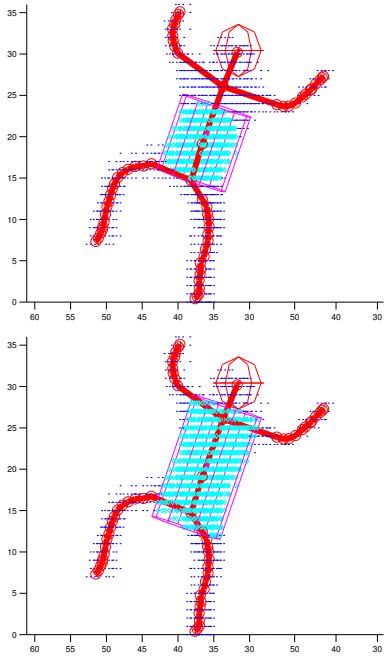

Figure 7: *Initial and final segmentation of torso points*

lower parts without additional assumptions or information. If the threshold is exceeded, the volume points of the limb are divided by a separating plane placed at the separating skeleton point. The upper and lower parts of the limbs have cylinder axes for line segments connecting the root limb, separating point, and leaf branch. The separating plane is oriented such that the angle formed between the cylinder axes is divided equally. The cylinder radius is determined by the average of the distance between "surface" volume points and the cylinder axis.

An example division of the volume points into rigid body parts is shown in (Fig.9).

After applying the rigid body fitting procedure for each individual frame, a second pass across the frames is used to correlate the rigid bodies across time. The second pass determines static radius and length values for the rigid bodies and consistent labels for each limb. The radius and length of each rigid body is determined from its average value in all frames in which produce distinct articulations. Limb values in postures without articulations are not included in the averaging. Limbs are provided with a consistent labeling, such as "arm1", using the sum of squares distance between rigid body locations in adjacent frames for correspondence. Semantic labels for "right" and "left" are not applied to the limbs.

The rigid bodies represented in world coordinates are converted into local coordinates with respect to

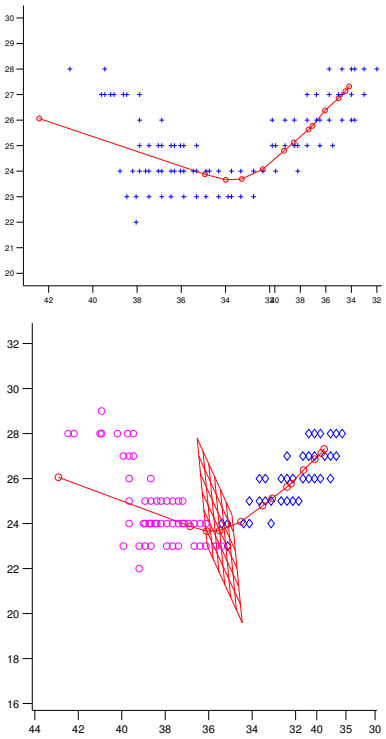

Figure 8: *Segmentation of an arm points into upper and lower parts.*

their parent in the kinematic hierarchy. More specifically, the torso is expressed in world coordinates, the head and upper limb bodies are expressed with respect to the torso, and the lower limb bodies are expressed with respect to their corresponding upper limb body. The origins of the torso and the head are located at the center of the body geometry. The origins of the upper and lower limb bodies are located at the limb roots and limb separating points, respectively.

The Z axis of each local coordinate system is defined by the cylinder axis of the rigid body. The positive direction of these axes points toward the head for the torso and outward from the torso for upper limb bodies, and outward from the separating point for lower limb bodies. The Z axis of the head is the vector difference of the upper most point of the torso cylinder axis from the center of the head.

The X axis of the torso is the cross product of the Z axis and the vector formed between the two shoulder points. Given the defined torso coordinate system, child bodies in the kinematics can determine their X axes by the cross product of X axis of their parent and their own Z axis. If these axes are parallel, an offset of the parent X axis is used in the cross product. Y axes can then be determined simply through the cross

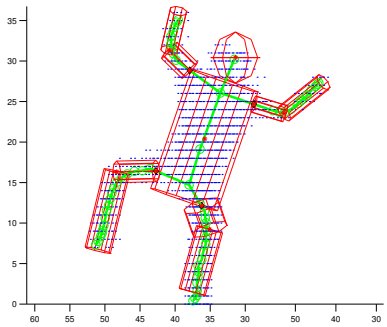

Figure 9: *The fitting of cylinders and spheres to a human volume*

product of it respective X and Z axes. Joint angles are then determined from the rotational transformation between each child body and its parent.

5 Results and Discussion

In this section, we discuss the implementation of our approach and present our results from capturing various kinematic motions. The entire implementation was developed and executed on a 350 MHz Pentium with 128 GB of memory. The voxel carving procedure was implemented in Microsoft Visual C++ for a $80 \times 80 \times 50$ grid of $50mm^3$ voxels. Images were taken from four Marshall V-1246-T CCD camera and captured by an Imagenation PXC200 framegrabber. Using the Intel Image Processing Library, volumes were captured in interactive-time, approx. 10 Hz.

Our voxel carving procedure was used to capture point volume sequences of a human marching, waving, and jumping jacks, shown in figure 10. These motions were selected because they exhibit the expected kinematic articulation. Each sequence consisted of approximately 50 frames.

The skeleton curve and kinematic posture estimation procedures were implemented in Matlab, as shown in figure 10. For Isomap, we an epsilon distance radius of $(50mm^3)^{1/2}$ produce an embedding that accentuates the underlying kinematics, but does not reduce dimensionality. For concentric spherical shells, the embedded volume is divided into 25 shells and the bounding box clustering threshold is set to $1/25^{th}$ of the diagonal of embedded volume bounding box. For all of the volume sequences, the estimated skeleton curves and kinematic postures accurately reflected the underlying posture of the captured human. The estimated kinematic motion of each sequence was exported to Biovision BVH motion capture files, specifying the Euler

angles for each posture. Each exported motion used as desired joint space trajectory for controlling a 20 DOF dynamically simulated humanoid torso, Adonis [7].

Our approach and implementation provide a useful system for markerless human motion capture, but has certain limitations. One such limitation is imposed by the quality of the volumes produced from the sensing human. Our current multi-camera system is rather unrefined. We use relatively low quality (but affordable) cameras, placed in a non-optimal configuration due to lab space constraints. Because image segmentation is not our focus, our background subtraction is a rough version of [4]. These factors manifest themselves in the sensed volume through erroneous shadow voxels on the floor and "ghost voxels". A ghost voxel is erroneously activated due to pixels viewing different objects that fuse in persepective projection. To account for these problems, we clean up the raw volume through offline procedures such as removing floor voxels and using the single largest connected component in the grid, in addition to manual clean up. The volume sensing has much room for accuracy and speed improvments.

Our current implementation also has a limitation the N number of volume points that can be processed. In order to perform nonlinear embeddings, Isomap performs an eigenvalue decomposition on an $N \times N$ matrix. If N exceeds roughly 3000 points, Matlab will not complete the Isomap procedure due to memory constrints. In addition, the execution of Isomap typically accounts for 90 percent of the processing in the volume to posture conversion. Even though the volume conversion process is not slow (60 to 90 seconds), we would like to make the procedure as fast as possible. We reduce the effect of these limitations in two ways. One, we can use a subset of the sensed volume points. Two, Isomap provides the use of landmarks in order to use an $M \times N$, where $M < N$.

An additional benefit to using our volume to posture conversion is potential to derive kinematic models and postures from volumes with arbitrary topologies. Furthermore, our approach can be extended to extract tree structured kinematics for volumes that indicate the presence of cycles.

6 Conclusion

We have presented a method for converting sequences of human point volumes sensed from calibrated multiple cameras into kinematic human motion. We introduced a new method for finding principal curves from volume data. These principal curves enabled us to classify volume points into rigid bodies and determine kinematic information about the input volume. We demonstrated the potential for uninstrumented robotic teleoperation by actuating motion produce by our system on a huanoid torso simulation.

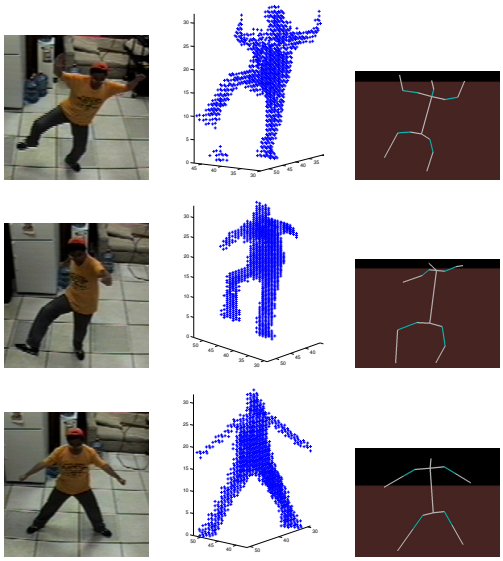

Figure 10: *Results from processing motions for waving, walking, jumping jacks (rows). The results are shown as a snapshot of the performing human, the sensed point volume, and the derived kinematic posture (columns).*

References

[1] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[2] Jean-Yves Bouguet. Camera calibration toolbox for matlab. http://www.vision.caltech.edu/bouguetj/calib_doc/index.html.

[3] Jonathan D. Cohen, Ming C. Lin, Dinesh Manocha, and Madhav K. Ponamgi. I-COLLIDE: An interactive and exact collision detection system for large-scale environments. In *Proceedings of the 1995 symposium on Interactive 3D graphics*, pages 189–196, 218. ACM Press, 1995.

[4] Alexandre R.J. Francois and Gérard G. Medioni. Adaptive color background modeling for real-time segmentation of video streams. In *Proceedings of the International on Imaging Science, Systems, and Technology*, pages 227–232, Las Vegas, Nevada, June 1999.

[5] T. Hastie and W. Stuetzle. Principal curves. *Journal of the American Statistical Association*, 84:502–516, 1989.

[6] Balazs Kegl, Adam Krzyzak, Tamas Linder, and Kenneth Zeger. Learning and design of principal curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(3):281–297, 2000.

[7] Maja J Matarić, Victor B. Zordan, and Z. Mason. Movement control methods for complex, dynamically simulated agents: Adonis dances the macarena. In *Autonomous Agents*, pages 317–324, 1998.

[8] Ivana Mikić, Mohan Trivedi, Edward Hunter, and Pamela Cosman. Articulated body posture estimation from multi-camera voxel data. In *IEEE International Conference on Computer Vision and Pattern Recognition*, Kauai, Hawaii, December 2001.

[9] Simon G Penny, Jeffrey Smith, and Andre Bernhardt. Traces: Wireless full body tracking in the cave. In *Ninth International Conference on Artificial Reality and Telexistence (ICAT'99)*, December 1999.

[10] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.