Skeleton-based Joints Position Detection

Martin Madaras, Michal Piovarči and Tomáš Kovačovský

madaras@sccg.sk
Comenius University Bratislava, Slovakia

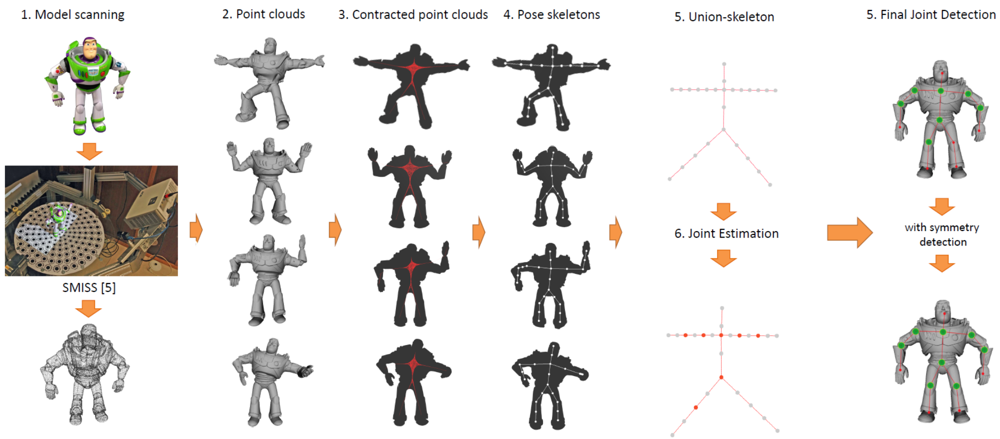

Figure 1: *Workflow of our system: from scanning of input articulated model to final detection of its joints.*

Abstract

We present a system for detection of joint positions in scans of articulated models. Our method is based purely on skeletons extracted from scanned point clouds of input models. First, skeletons are extracted from scans and then an estimation of possible matches between skeletons is performed. The matches are evaluated and sorted out. The whole matching process is fully automatic, but some user-driven suggestions can be included. Finally, we pick the best matching of skeletons and create a union-skeleton containing all the nodes from all the skeletons. We find nodes in the union-skeleton with rotation changes higher than the predefined threshold. We take these nodes as joints and visualize them in original scans.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, shading, shadowing, and texture

1. Overview

Our system is composed of three interlocking steps. Firstly, we scan an input articulated model in different poses (Section 2). The result of the scanning is a set of point clouds. Secondly, we extract skeletons from these point clouds (Section 3). The resulting skeletons are matched using our graph-based heuristics (Section 4). The skeletons are matched automatically and a union-skeleton is created. However, the user can change the matching of some segments, if desired. The union-skeleton consists of all the nodes from all the extracted skeletons. Finally, the positions of joints are detected from the union-skeleton and its comparison to other skeletons. The nodes with the biggest rotation changes are detected as real joints (Section 5). Workflow of our system can be seen in Figure 1. Note that all the existing video and kinect-based approaches as [SSK*13] work on human-like figures only. They are based on fitting of skeleton templates into a set of images. Our approach is general and works for models with arbitrary topology and skeleton branching.

2. Model Scanning

To obtain a complete high resolution point-clouds of articulated figures, we built our own 3D scanning system based on our structured light optical 3D scanner SMISS [Kov12], which uses projector-camera setup to retrieve a 3D surface information using triangulation principle. Stereo correspondence-matching is done using Gray and sinusoidal coded patterns to grant sub-pixel accuracy. We have further enhanced the system with motorised glass table for object rotation. The scanner can be positioned on arbitrary point of a sphere with a predefined radius with respect to a center of a object using a motorized arm. Such a construction allows to capture several single view 3D scans of the target object from various viewpoints. The object is laid on our newly designed pose-estimation marker field of multiple circles with binary coded positions (see Figure 1). In every view of the object, multiple circles from the background are localised and a pose estimation is computed from image to world space correspondence and the known calibration parameters (world coordinates are encoded as two 16 bits floats), so that individual scans have the same coordinate space independent from scanner position. The initially aligned scans are then aligned more precisely using ICP algorithm, to form complete point clouds of the objects.

3. Skeleton Extraction

For skeleton extraction we use a modified version of Au's algorithm [ATC*08] extended to point clouds in which we use a construction of local Delaunay's triangulations for Laplacian estimation. Similar solution is presented in [CTO*10], but the authors computed Euclidian distance between samples for construction of skeleton from the contracted point cloud. Such a distance grouping yields into problems when the structures in different skeleton branches are closer than point samples in the neighborhood of the vertex. Therefore, to construct the final skeleton, a simplification of a triangulation instead of Euclidian distance is used. This triangulation is constructed by composing local one-ring Delaunay's triangulations into the global triangulation.

4. Skeleton Matching

We start by removing connection nodes from the input skeletons so they contain only branch and leaf nodes. By removing the connection nodes, we lose the original information. To rectify this, we extend the edges of the input skeleton such that they contain data that help during the matching phase. Currently, we store the length of the removed path and the number of nodes that were on the path. In next phase, we are looking for matching between two skeletons using graph representation. Therefore, we convert skeletons to graphs with undirected edges. We use simple backtracking to find all the possible solutions. We impose two constraints to reduce the searching area. First constraint is that each newly matched node from the first skeleton must have the same neighbors as corresponding node in the second skeleton. The second constraint is that a node can only match onto a node with the same or higher number of neighbors. Two-pass filtering is applied on the rated solutions. In the first pass, the matching is penalized for unequal number of vertices on the path. Solutions are then sorted from best to worst. We pick only the solutions, whichs raiting is within a threshold of tolerance from the best solution. In the second pass, we sort these solutions again with different rating which measures Euclidean distance between matched nodes in skeletons.

5. Joints Estimation

We merge all the matched skeleton segments into one union-skeleton. For every union-skeleton node, we measure the change in rotation between original skeleton pose and the matched superskeleton. Nodes where rotation changes are higher than predefined threshold are detected as joints. Finally, a skeleton symmetry is used to detect joints in symmetrical body parts of the model.

6. Applications and Future Work

With existing approaches for 3D printing of articulated models [BBJP12] and detection of joints in combination with automatic extraction of skinning weights, our approach can be used for automatic cloning of articulated toys. Furthermore, we would like to extend this approach to humans. This could be used for detection of joint position and calculating length of human bones. Such an extraction of length of human bones could be used for biometry, motion capture or other analysis of human body.

References

[ATC*08] Au O. K.-C., Tai C.-L., Chu H.-K., Cohen-Or D., Lee T.-Y.: Skeleton extraction by mesh contraction. In *SIGGRAPH '08: ACM SIGGRAPH 2008 papers* (2008), pp. 1–10. 2

[BBJP12] Bächer M., Bickel B., James D. L., Pfister H.: Fabricating articulated characters from skinned meshes. *ACM Trans. Graph. (Proc. SIGGRAPH) 31*, 4 (2012). 2

[CTO*10] Cao J., Tagliasacchi A., Olson M., Zhang H., Su Z.: Point cloud skeletons via Laplacian based contraction. In *Proceedings of the 2010 SMI Conference* (2010), pp. 187–197. 2

[Kov12] Kovačovský T.: Scalable multifunctional indoor scanning system. In *Bulletin of the ACM Slova.* (2012), pp. 47–48. 2

[SSK*13] Shotton J., Sharp T., Kipman A., Fitzgibbon A., Finocchio M., Blake A., Cook M., Moore R.: Real-time human pose recognition in parts from single depth images. *Commun. ACM 56*, 1 (Jan. 2013), 116–124. 2