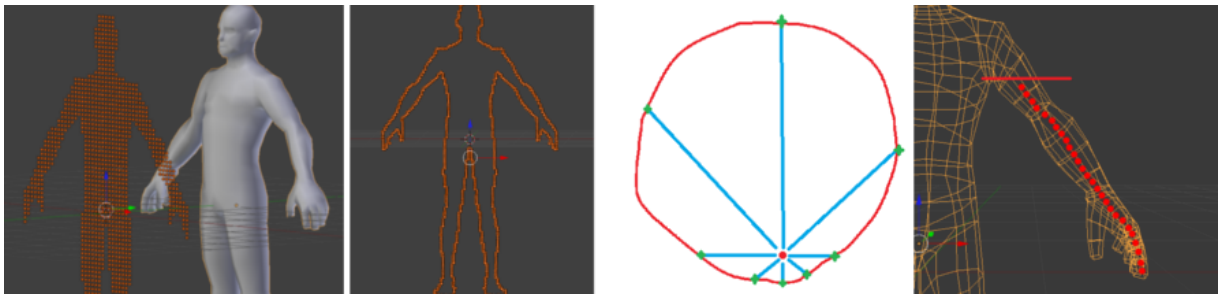
Raycast based auto-rigging method for humanoid meshes

Romain Lopez, Christophe Poirel of ESGI Paris



(a) (b) (c) (d)

Figure 1: *(a) Mesh "silhouette", (b) Silhouette outline, (c) Raycasting inside a limb slice, (d) Arm slices centers.*

1 Introduction

In character animation, skeletons are used to animate meshes. Those skeletons are inserted inside the mesh after being drawn. This insertion, called rigging, is time consuming since it must be done for each mesh.

Autorigging is the automation of this process. In this work, we propose an automation which requires no human intervention and will identify and place each bone of the skeleton inside a mesh.

Other works were done on the same subject : Pinocchio which uses mesh discretization [Baran 2007] and Frankenrigs which is based on a body parts database [Miller 2010] .

2 Body map

In the first part, we will try to build a body map : we will scan the mesh and try to identify all the limbs. The limbs we are looking for are based on the bones we have in the skeleton : arms, legs, body and head.

The first step is to reduce the complexity of our 3D point cloud (the mesh). In order to do so, we are going to make a serie of frontal ray casts. Each raycast will be sent with a regular spacing. We will keep all intersection points with the mesh and set a common depth for them. By the end of this first step, we will then have what we called the "silhouette" of the mesh (Figure 1a), which looks like a 2D point cloud projection, but with a regular spacing between each point. Since we know the spacing between each point, we can easily keep only the outline points of the cloud (Figure 1b). The result still is recognizable as a human shape, and less complex than the 3D point cloud.

What we want to do next is to set these points in a way that will allow the program to read the silhouette. Just like if you started to draw this figure with a pen, you would for example start by drawing one point in the head, then the next point close to it... until you go back to the first point. You will then have a navigation path, which starts at the first point you drew and goes all around the silhouette.

We will be building three paths, one arch that goes inside the legs, two others that go on the side of the mesh. This can be done, because we used a regular spacing, and we know the distance between two neighbor points. Using these tracks and knowing the standard rigging positions, we will be able to determine each limb.

SIGGRAPH 2013, July 21 – 25, 2013, Anaheim, California.

ACM 978-1-4503-2261-4/13/07

We will be doing a specific algorithm for the legs, arms, body and head. At the end of the first part, we will have our body map ready.

3 Bone placement

The bone placement consists in finding the best location for each joint of the skeleton. We we will use the body map to find entry points in the mesh for each limb. These entry points will be used as start locations and stop locations for the following algorithm.

We start with the arm entry point. The armpit height will be the height limit, i.e. the stop condition of the algorithm. From the entry point inside the arm, we do a 360 degree raycast in the horizontal plane (Figure 1c). These rays will hit the mesh from inside and we keep the closest intersection points. A polygon is made from these intersection points. It represents a simplified slice of the arm. Now the goal is to find the center of this slice. To do so, we will find the barycenter of the polygon by applying a weight on each point depending on the distance to its two neighbors. This way the barycenter's position will not be affected by the local points density. From this barycenter, we will go down the arm. Again, at each slice, we do a raycast to find the next barycenter. The algorithm stops when the next point is outside the mesh. Then we start again at the arm entry point, but this time we go upward. The algorithm stops when it reaches the armpit height (Figure 1d). The barycenters found at each step are candidates for the joint positions. We use other entry points from the body map to find the legs and the spine.

The last step is to position the armature bones. From the barycenters, we must find the best candidates for the different joints locations.To do so, we use the armature skeleton information. For each bone, we take the ratio between its length and the total length of the limb bones. Then, inside the mesh, we sum the distance between each barycenter to get the total length of the correponding limb. Now we can use the bones ratios to find which barycenters will be the locations of the joints.

References

BARAN, I. AND POPOVIC, J., 2007. Automatic rigging and animation of 3d character. Proceedings of ACM SIGGRAPH 2007.

MILLER, C., ARIKAN, O., AND FUSSELL, D., 2010. Frankenrigs: building character rigs from multiple sources. Proceedings of ACM SIGGRAPH 2010.