Chapter 15
Designing Non-photorealistic Animation Collages

In this chapter, we present a method to automatically transform mesh animations into animation collages, i.e. a new non-photorealistic rendering style for animations. By automatically decomposing input animations and fitting a shape from the database into each segment, our algorithm creates a new rendering style. It has many applications in arts, non-photorealistic rendering, and animated movie productions.

An animation collage is a complete reassembly of the original animation in a new abstract visual style that imitates the spatio-temporal shape and deformation of the input. Many researchers in computer graphics have been inspired by the idea to develop algorithms that enable the computer and even inexperienced users to reproduce the look of certain styles of visual arts, such as collages. Kim et al. [10] develop a system that can automatically turn arbitrary photographs into collage mosaics that comprise of an arrangement of elementary image tiles. Rother et al. [14] automatically arrange and blend photographs from a database into a perceptually pleasing way. Gal et al. [5] show results of a method to approximate static 3D shapes with other meshes, but they do not handle the general case of mesh animations.

In contrast, this chapter presents a method allowing a computer artist to automatically convert his/her favorite mesh animation into a moving assembly of 3D shape primitives in a database. This so-called animation collage is glued together in such a way that it approximates the sequence of shapes from the original mesh animation, while deforming in the same spatio-temporally consistent way as the original. While our method can fully-automatically build moving collages, it also purposefully gives the artist the possibility to post-process and fine-tune the results according to his/her imagination.

An overview of our system is presented in Sect. 15.1. It first automatically decomposes the input mesh animation into moving approximately rigid volume segments, henceforth called animation cells, Sect. 15.2 and Sect. 15.3. This decomposition is learned from the moving input meshes by means of a spectral clustering approach, Sect. 14.2. Thereafter, it employs a spatio-temporal matching criterion that
analyzes the motion and deformation of each animation cell and finds a shape primitive in the database that best approximates its time-varying geometry, Sect. 15.4. Shape primitives and cells are spatio-temporally aligned, and the fitted shapes are moved and deformed according to the deformation of the cells, Sect. 15.5. Since it is also our goal to develop new algorithmic recipes for a novel artistic tool, an animator can influence the final result at all stages of the processing pipeline, Sect. 15.6.

The main contribution of this chapter is a system [15] to

- automatically transform mesh animations, which are created directly by animators or captured using our previous performance capture methods, into animation collages.

Our software prototype is easy to use and allows even untrained users to create very aesthetic collages. Therefore, our system is an interesting add-on to the graphics artist’s toolbox, with many applications in visual arts, non-photorealistic rendering, and productions of games and cartoons.

15.1 Overview

As in Chapter [14], the input to our algorithm is an animated mesh sequence comprising of \( N \) frames, represented by a mesh model \( \mathcal{M}_{\text{tri}} = (V_{\text{tri}}, T_{\text{tri}}) \) and position data \( p_i(v_j) = (x_j, y_j, z_j)_t \) for each vertex \( v_j \in V_{\text{tri}} \) at all time steps \( t \). The coordinate sets \( P_t = \{ p_t(v_j) \} \) together with \( \mathcal{M}_{\text{tri}} \) describe a time-varying surface. The second input element is a database of \( K \) static shapes, each being represented as a textured triangle mesh.

The first step in our pipeline is the motion-decomposition of the mesh. To this end, we employ the method described in Sect. 14.2 that analyzes the motion of the mesh and delivers contingent triangle patches representing approximately rigid elements, as shown in Fig. 15.1(a). To enable the fitting of shape primitives, we transform the rigid surface elements into approximately rigid volume cells, so-called animation cells. To this end, a sequence of medial axis meshes is computed from the animation which, in conjunction with the previously identified rigid surface segments, is used to create these closed volume cells, Sect. 15.3 and Figs. 15.1(b),(c). Once the animation cells have been identified, we automatically fit to each of them a shape primitive from the database, Sect. 15.4. The final moving collages are generated by deforming the fitted shapes according to the transformation of their respective animation cells, Fig. 15.1(d). To achieve this, we generate spatio-temporally consistent offset meshes from the animation cells that drive the shape primitives’ deformations, Sect. 15.5.

15.2 Rigid Body Segmentation

The first step in our pipeline segments the input animation given by \( \mathcal{M}_{\text{tri}} \) and \( P_t \) into spatially coherent triangle patches that undergo approximately the same rigid transformations over time. Our motivation for decomposing the mesh into approximately