Multi-resolution Clustering for Enhanced Elastic Behavior in Clustered Shape Matching (Supplemental Material)

Aditya Viswanathan Kaliappan
adityak1@umbc.edu
University of Maryland, Baltimore County (UMBC)
Baltimore, Maryland, USA

Adam W. Bargteil
adamb@umbc.edu
University of Maryland, Baltimore County (UMBC)
Baltimore, Maryland, USA

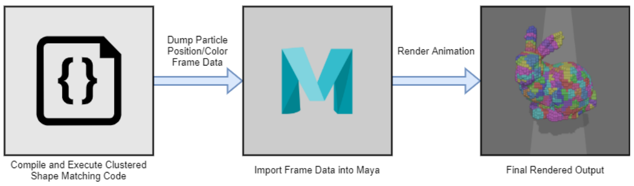

Figure 1: Rendering Pipeline.

1 RENDERING PIPELINE

A high-level graphical overview of the rendering pipeline is presented in Figure 1. All animation results presented in this paper, the main paper, and supplemental video are rendered using Autodesk® Maya® 2020[1] with the Maya Hardware 2.0 renderer. Our code dumps the particle positions and associated nearest cluster colors per frame as .txt files. This data is then imported into Maya® as a set of particle objects using a custom Maya Embedded Language (MEL)[2] script. Alternatively, we skinned the set of particles per frame into a smoothed mesh before importing them into Maya.

2 ANIMATION AND WEIGHT CONFIGURATIONS

In this section, we present the specific parameters and weights used for each of the animation examples in the main paper. It is important to note that the normalized weights in all the weight tables are presented in order from finest to coarsest resolution level.

2.1 Falling Ball

Table 1 contains the parameter configurations used for the falling ball example in the main paper. Figure 2 shows the nearest cluster per particle for each resolution level. Table 2 shows the normalized weights used per resolution level with this example.

[1]https://www.autodesk.com/products/maya/overview
[2]http://help.autodesk.com/cloudhelp/2020/ENU/Maya-Tech-Docs/Commands/index.html

MIG '20, October 16–18, 2020, Virtual Event, SC, USA

ACM ISBN 978-1-4503-8171-0/20/10…$15.00
https://doi.org/10.1145/3424636.3426902

Parameters		$\lvert\mathcal{P}\rvert$	5,098 particles
		α	0.3
		k_d	0.18
		$gravity$	-9.8m/s^2
		m	2.6
Parameters by Level	Level 0	N	101 clusters
		r	0.125
	Level 1	N	12 clusters
		r	0.325
	Level 2	N	1 cluster
		r	0.845

Table 1: Parameters Used For Falling Ball Animation

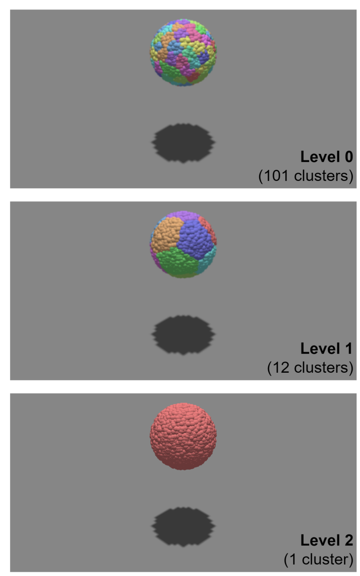

Figure 2: Clustering Ball by Level.

Animation	Normalized Weights
Uniform	[0.333333 0.333333 0.333333]
Linear, $\epsilon = 0.1$ (favor coarser levels)	[0.030303 0.333333 0.636364]
Gaussian (favor finer levels)	[0.574097 0.348207 0.0776956]
Gaussian (favor coarser levels)	[0.0776956 0.348207 0.574097]
Polynomial, $b = 5, c = 4$ (favor coarser levels)	[0.00247419 0.125115 0.872411]
Manual	[0.1 0.8 0.1]

Table 2: Weights For Falling Ball Animation

Parameters		$\|\mathcal{P}\|$	1,331 particles
		α	0.75
		k_d	0.0
		$gravity$	$0.0 m/s^2$
		m	2.0
Parameters by Level	Level 0	N	66 clusters
		r	0.25
	Level 1	N	8 clusters
		r	0.5
	Level 2	N	1 cluster
		r	1.0

Table 3: Parameters Used For Hovering Cube Animation

Animation	Normalized Weights
Uniform	[0.333333 0.333333 0.333333]
Linear, $\epsilon = 0.1$ (favor coarser levels)	[0.030303 0.333333 0.636364]
Linear, $\epsilon = 0.1$ (favor finer levels)	[0.636364 0.333333 0.030303]
Gaussian (favor finer levels)	[0.574097 0.348207 0.0776956]
Polynomial, $b = 8, c = 3$ (favor coarser levels)	[0.00328587 0.161981 0.834733]
Polynomial, $b = 8, c = 3$ (favor finer levels)	[0.834733 0.161981 0.00328587]

Table 4: Weights For Hovering Cube Animation

Parameters		$\|\mathcal{P}\|$	16,510 particles
		α	0.5
		k_d	0.5
		$gravity$	$-9.8 m/s^2$
		m	2.07
Parameters by Level	Level 0	N	330 clusters
		r	0.275
	Level 1	N	41 clusters
		r	0.56925
	Level 2	N	5 clusters
		r	1.1783475
	Level 3	N	1 cluster
		r	2.439179325

Table 5: Parameters Used For Falling Bunny Animation

2.2 Hovering Cube

Table 3 contains the parameter configurations used for the hovering cube example in the main paper. Table 4 shows the normalized weights used per resolution level with this example.

2.3 Falling Bunny

Table 5 contains the parameter configurations used for the falling bunny example in the main paper. Table 6 shows the normalized weights used per resolution level with this example.

2.4 Ball Colliding Into Wall

Tables 7 and 8 contain the parameter configurations used for the ball colliding into wall example in the main paper. Table 9 shows the normalized weights used per resolution level with this example.

Animation	Normalized Weights
Linear, $\epsilon = 0.01$ (favor finer levels)	[0.498344 0.332781 0.167219 0.00165563]
Linear, $\epsilon = 1$ (favor finer levels)	[0.4 0.3 0.2 0.1]
Linear, $\epsilon = 100$ (favor finer levels)	[0.253695 0.251232 0.248768 0.246305]
Polynomial, $b = 10, c = 0$ (favor finer levels)	[0.25 0.25 0.25 0.25]
Polynomial, $b = 10, c = 1$ (favor finer levels)	[0.447368 0.315789 0.184211 0.0526316]
Polynomial, $b = 10, c = 2$ (favor finer levels)	[0.59465 0.296296 0.100823 0.00823045]
Polynomial, $b = 10, c = 3$ (favor finer levels)	[0.70266 0.24714 0.0490561 0.00114416]

Table 6: Weights For Falling Bunny Animation

Parameters		$\|\mathcal{P}\|$	1,000 particles
		α	1.25
		k_d	0.1
		$gravity$	$0.0 m/s^2$
		m	1.75
Parameters by Level	Level 0	N	66 clusters
		r	0.15
	Level 1	N	8 clusters
		r	0.2625
	Level 2	N	1 cluster
		r	0.459375

Table 7: Parameters Used For Ball in Ball Colliding Into Wall Animation

Parameters		$\|\mathcal{P}\|$	1,000 particles
		α	1.25
		k_d	0.1
		$gravity$	$0.0 m/s^2$
		m	2.0
Parameters by Level	Level 0	N	66 clusters
		r	0.15
	Level 1	N	8 clusters
		r	0.3
	Level 2	N	1 cluster
		r	0.6

Table 8: Parameters Used For Wall in Ball Colliding Into Wall Animation

Animation	Normalized Weights
Manual Configuration #1	Ball: [0.5 0.3 0.2], Wall: [1 0 0]
Manual Configuration #2	Ball: [0.5 0.3 0.2], Wall: [0.25 0.7 0.05]
Manual Configuration #3	Ball: [0 0.1 0.9], Wall: [0 0.2 0.8]

Table 9: Weights For Ball Colliding Into Wall Animation

2.5 Three Bunnies Collide

Table 10 contains the parameter configurations used for the three bunnies collide example in the main paper. Table 11 shows the normalized weights used per resolution level with this example.

Parameters		$\lvert\mathcal{P}\rvert$	4,131 particles
		α	varies
		k_d	0.1
		$gravity$	$0.0 m/s^2$
		m	2.25
Parameters by Level	Level 0	N	206 clusters
		r	0.125
	Level 1	N	25 clusters
		r	0.28125
	Level 2	N	3 clusters
		r	0.6328125
	Level 3	N	1 cluster
		r	1.423828125

Table 10: Parameters Used Per Bunny in Three Bunnies Collide Animation

Animation	Normalized Weights
Gaussian & Polynomial	Left/Right: [0.570459 0.346001 0.0772032 0.00633722] Middle: [0.00014999 0.0225078 0.194387 0.782956]
Uniform	All: [0.25 0.25 0.25 0.25]

Table 11: Weights For Three Bunnies Collide Animation

Parameters		$\lvert\mathcal{P}\rvert$	5,098 particles
		α	0.3
		k_d	0.3
		$gravity$	$-5.0 m/s^2$
		m	2.6
Parameters by Level	Level 0	N	101 clusters
		r	0.1
	Level 1	N	12 clusters
		r	0.26
	Level 2	N	1 cluster
		r	0.676

Table 12: Parameters Used For Plinko Animation

Animation	Normalized Weights
Manual Configuration #1	[0.05 0.45 0.5]
Manual Configuration #2	[0.2 0.6 0.2]
Manual Configuration #3	[0.5 0.25 0.25]

Table 13: Weights For Plinko Animation

2.6 Plinko

Table 12 contains the parameter configurations used for the plinko example in the main paper. Table 13 shows the normalized weights used per resolution level with this example.

2.7 Armadillo

Table 14 contains the parameter configurations used for the armadillo example in the main paper. Figure 3 shows the nearest cluster per particle for each resolution level. Table 15 shows the normalized weights used per resolution level with this example.

Parameters		$\lvert\mathcal{P}\rvert$	124,631 particles
		α	1.25
		k_d	0.25
		$gravity$	$-5.0 m/s^2$
		m	2.125
Parameters by Level	Level 0	N	8192 clusters
		r	0.085
	Level 1	N	1024 clusters
		r	0.180625
	Level 2	N	128 clusters
		r	0.383828125
	Level 3	N	16 clusters
		r	0.815634765625
	Level 4	N	2 clusters
		r	1.733223876953125
	Level 5	N	1 cluster
		r	3.683100738525390625

Table 14: Parameters Used in Armadillo Animation

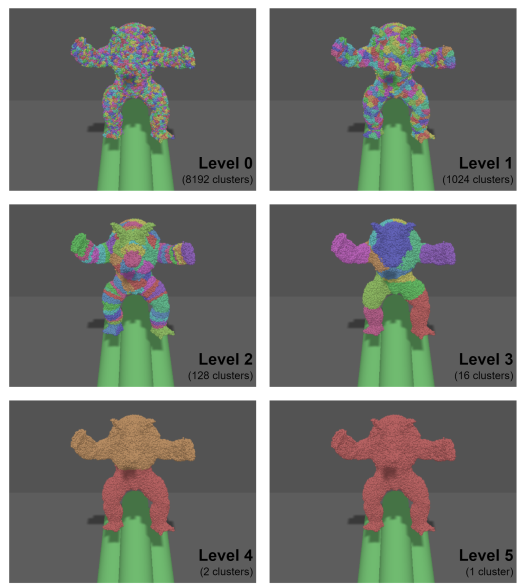

Figure 3: Clustering Armadillo by Level.

Animation	Normalized Weights
Linear, $\epsilon = 0.1$ (favor finer levels)	[0.326923 0.262821 0.198718 0.134615 0.0705128 0.00641026]
Manual	[0.85 0.01 0.01 0.01 0.06 0.06]

Table 15: Weights For Armadillo Animation