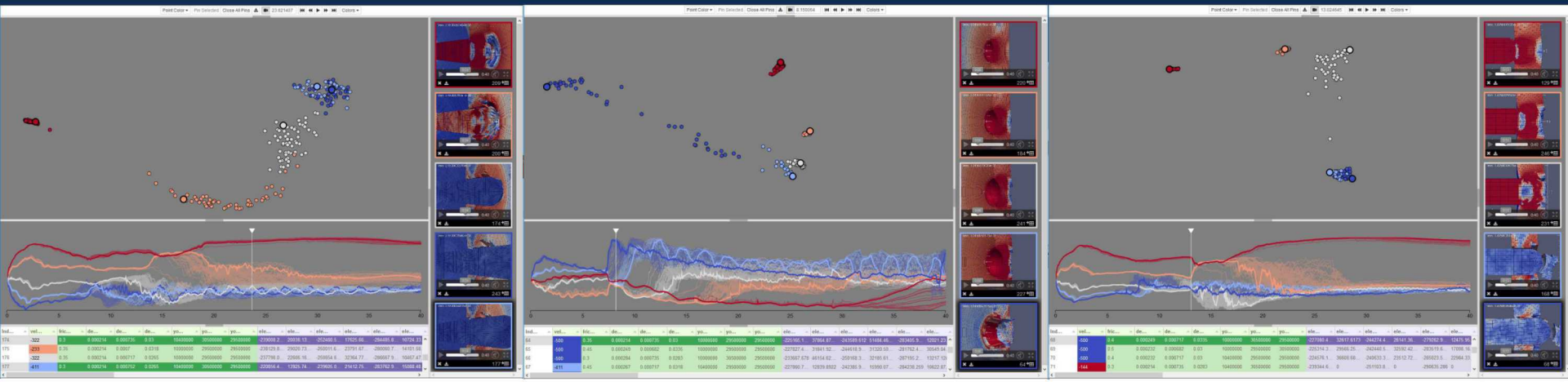
SAND2019-7044PE



Slycat™ VideoSwarm Update

June 2019

Patricia Crossno, Ph.D.

U.S. DEPARTMENT OF ENERGY

National Nuclear Security Administration

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

SAND2018-5807 PE

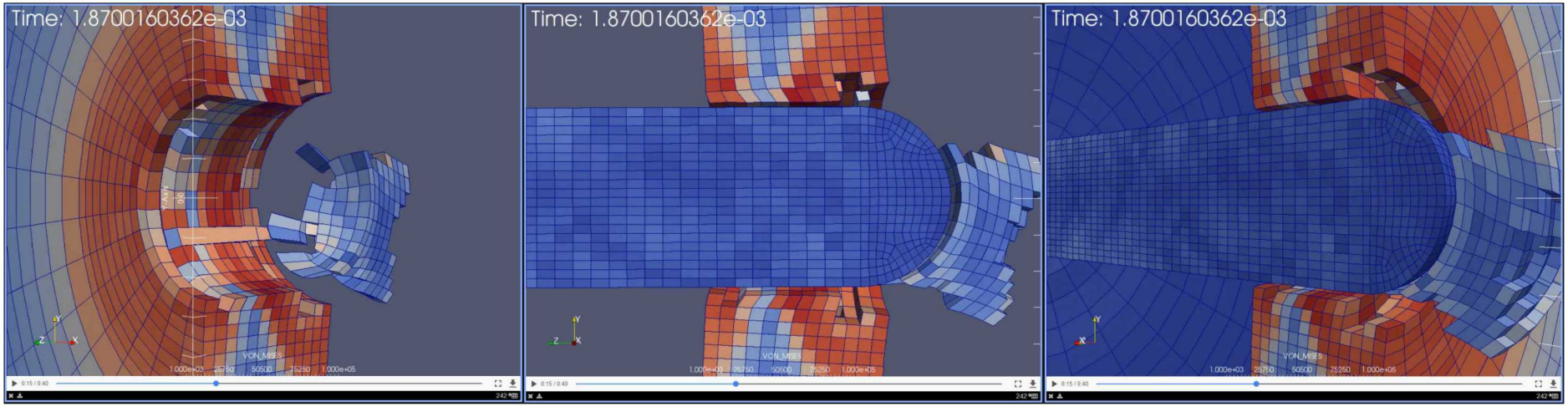
Video Ensembles

Sandia National Laboratories

- Simulations increasingly output video results
- Videos show physics models changing in time
- Directly viewing all videos doesn't scale
  - Sequential viewing
    - Viewing time required for a 30 sec video: 1000 runs = 8.3 hours (non-stop)
    - Cognitive limitations: unable to retain details or remember which video had which feature
    - Comparison: need to examine differences side-by-side
  - Concurrent viewing
    - Screen real estate limited: can't view all videos at full size at once. Thumbnails lose details and scrolling does not provide side-by-side comparison
    - Viewer focus: even if all videos fit on screen, user focus cannot be everywhere at once

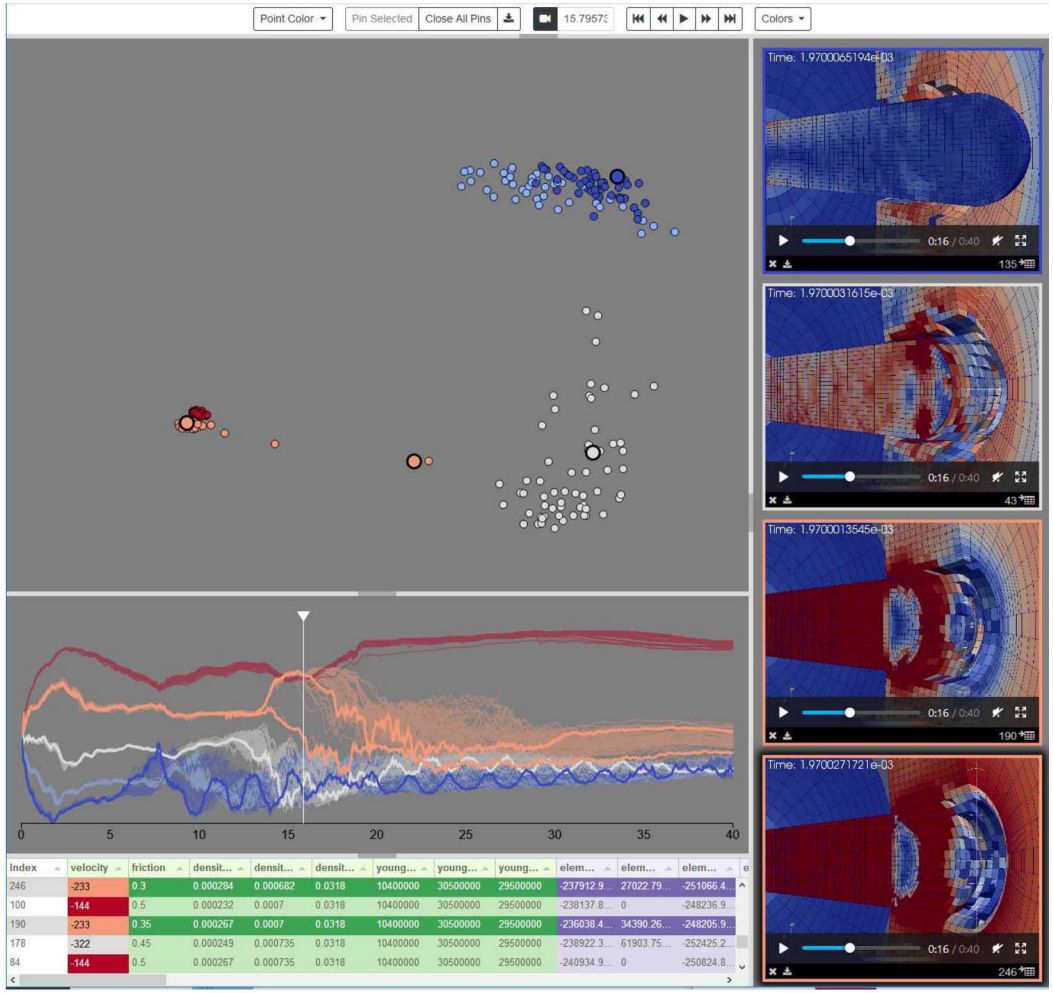
Video Ensemble Data

- Generated using Sierra/SolidMechanics
- Material fracturing problem, punch impacting plate
- 250 runs, each with:
  - 3 videos from different viewpoints
  - Each video with 1000 frames
- Frames were rendered using Von Mises stress to color cells



Video Analysis Tasks

- Identify clusters
- Find anomalies/outliers
- Discover events/discontinuities
- Reveal temporal behaviors
- Discover patterns between clusters and input values
- Synchronously compare exemplar videos



Video Ensemble Analysis Challenges

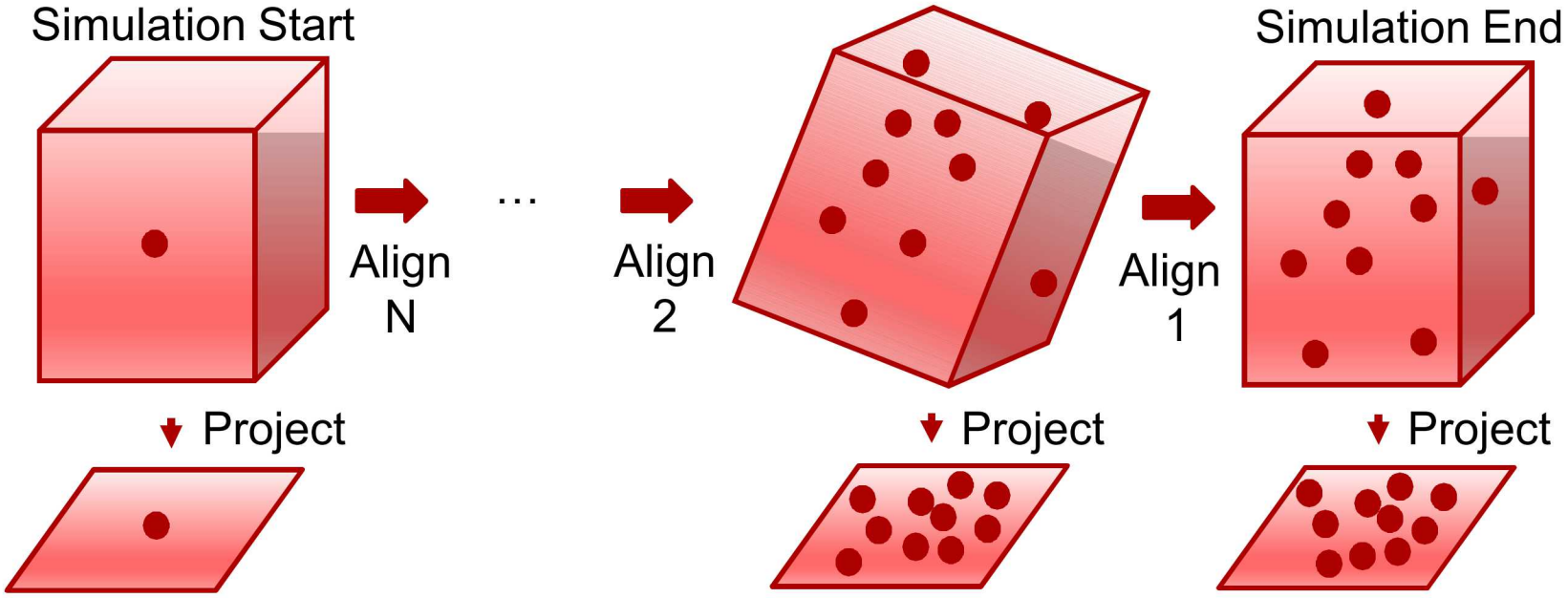
Sandia National Laboratories

- **Temporal** Multi-dimensional Scaling (MDS)
  - Time trajectories plotted as the first coordinate of MDS versus time.
  - Scatterplot coordinates are first two MDS coordinates (single time).
  - Calculations performed per frame
  - Need shared eigenvectors across time for coordinate space
  - Choice of frame as basis for alignment can bias results
  - Largest eigenvalues λ_1, λ_2 may be equal, so eigenvector ranking can flip between frames
- Analysis Performed on Images
  - Using RGB values as proxy for variable values
  - Disable auto-scaling to keep color meanings constant
  - Need value range for entire ensemble prior to image generation
  - Large dynamic range is problematic

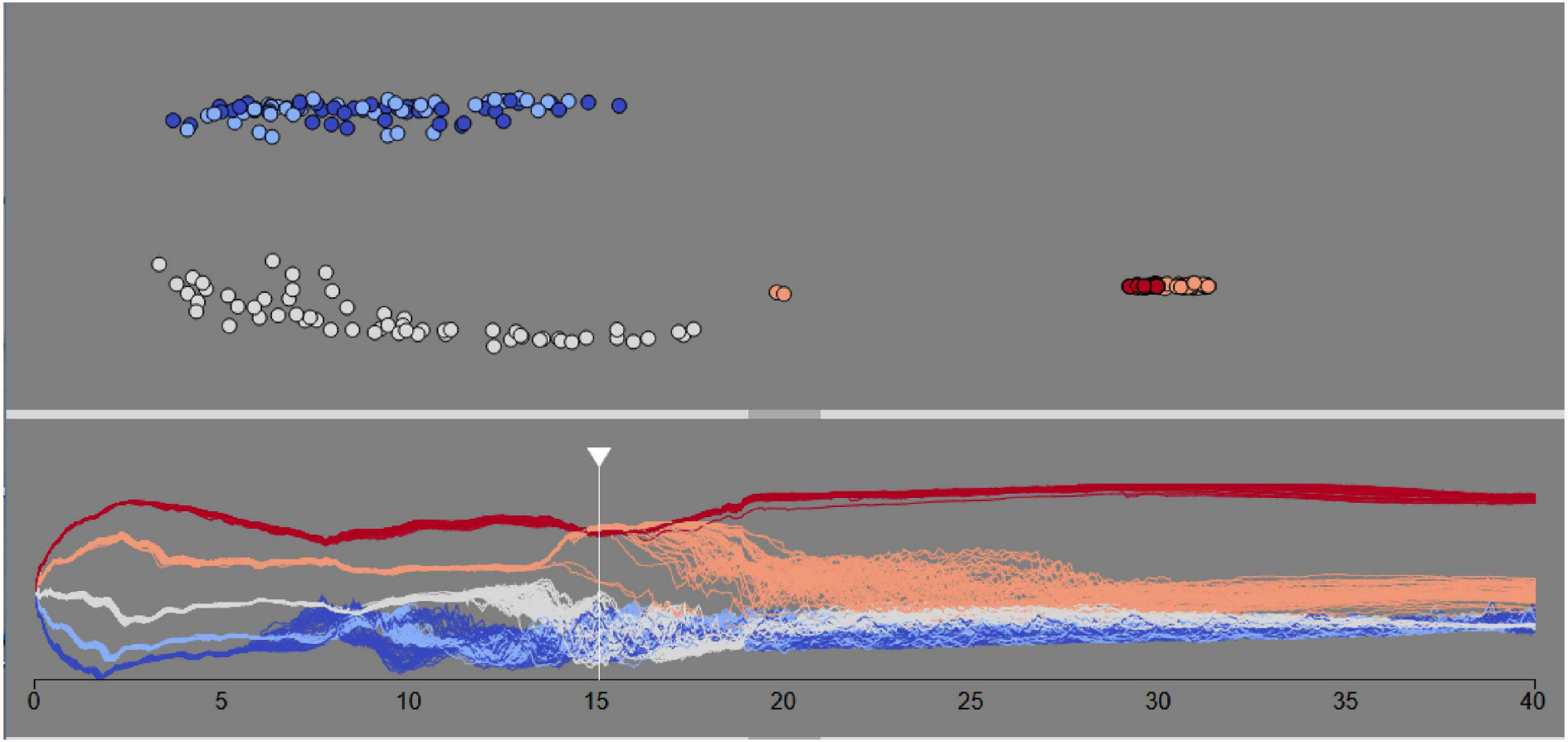
Multidimensional Scaling (MDS)

MDS calculation:

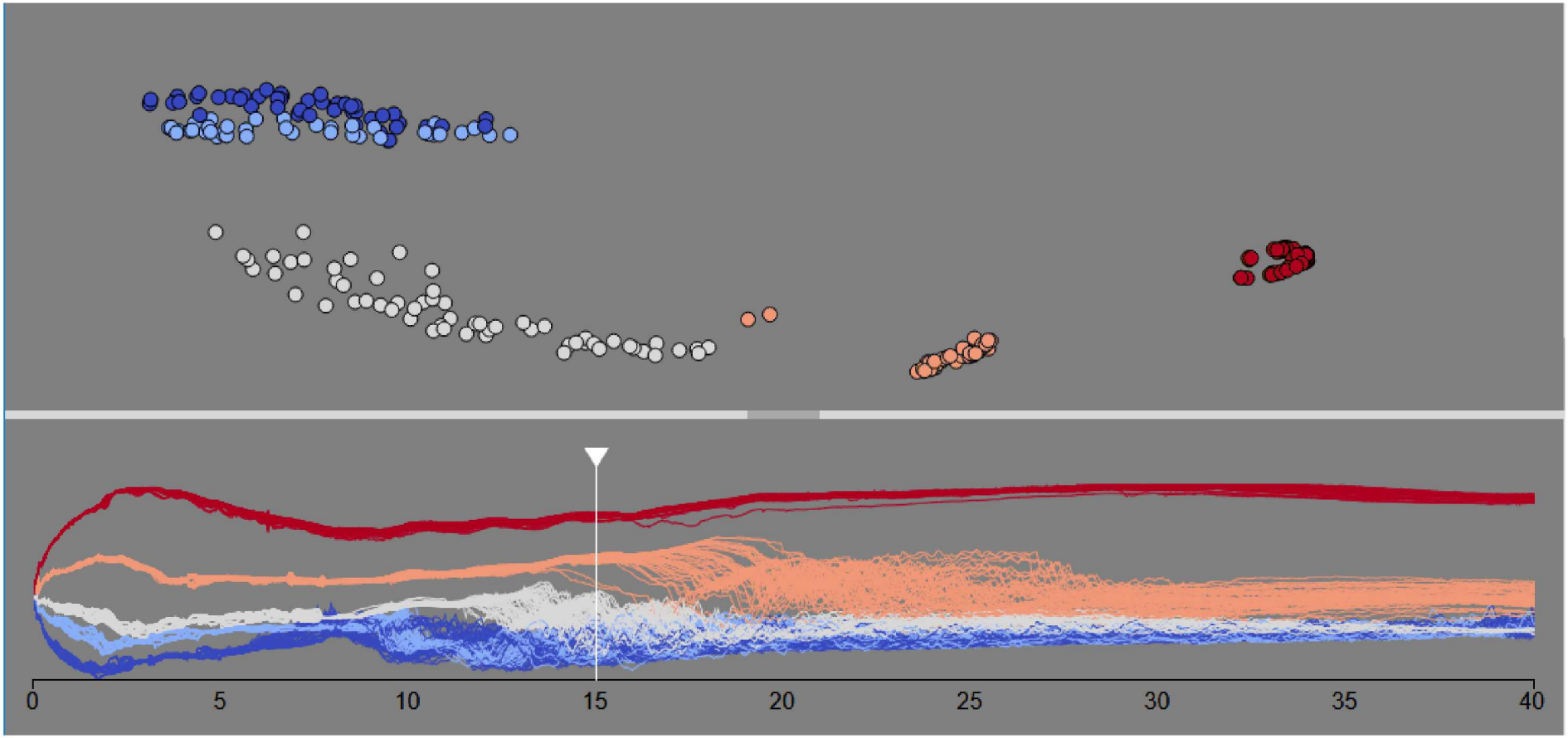
- Compute pair-wise distances between each frame of each video
- Double-center the distance matrix
- Compute the eigenvalue decomposition
- Align MDS visualization per frame using optimal orthogonal transform
- Align frames sequentially from last frame to first frame using *k* dims



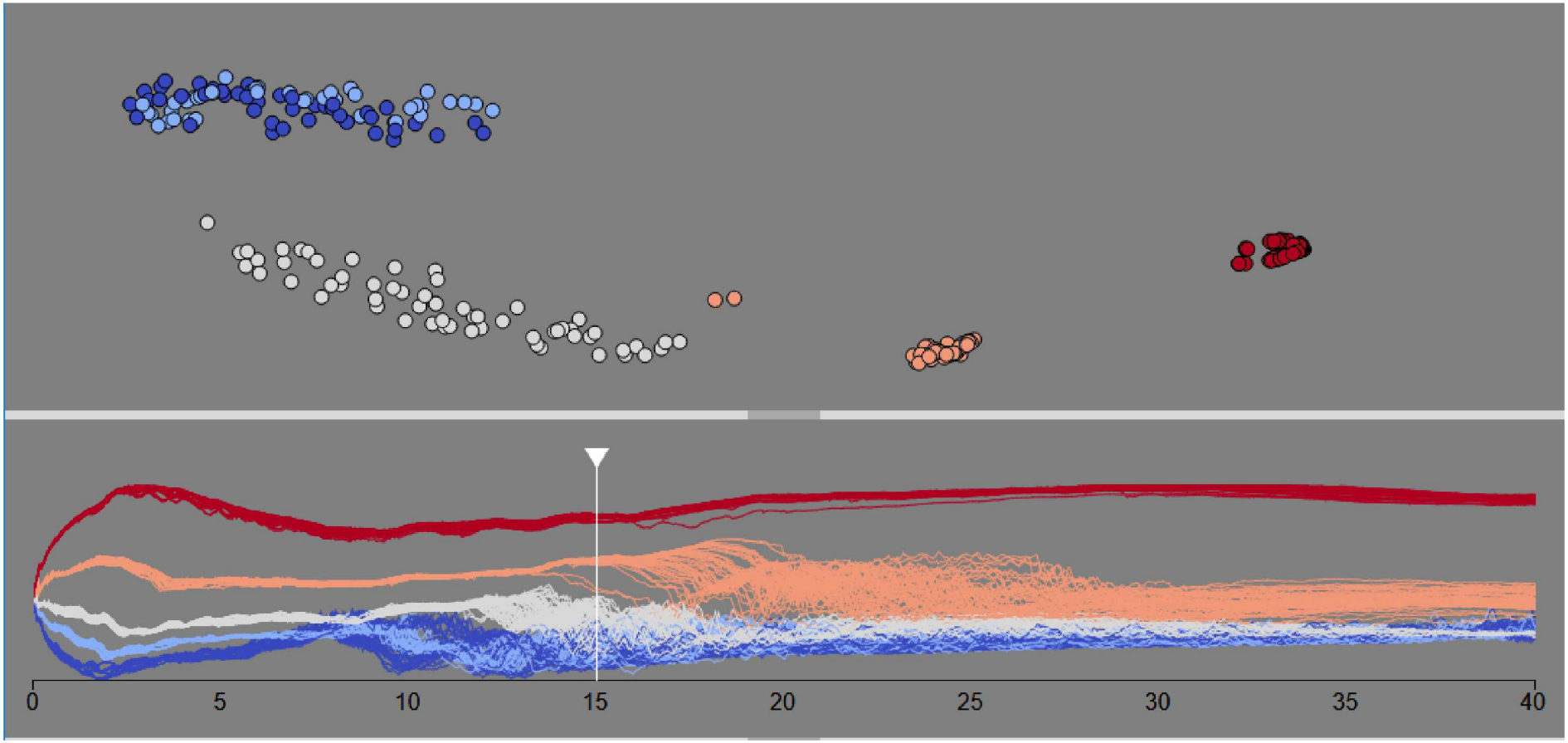
Dimension 2 Results



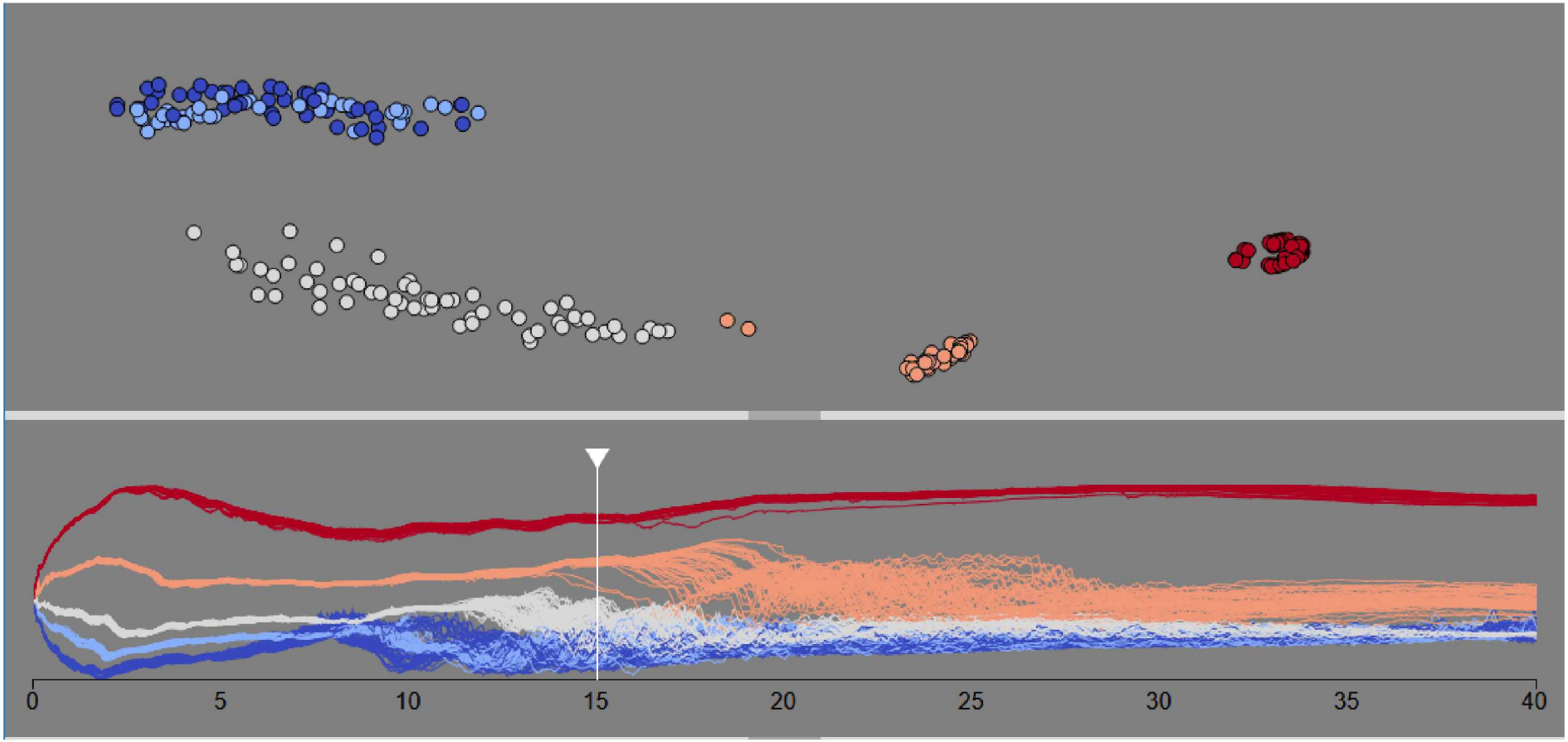
Dimension 10 Results



Dimension 20 Results

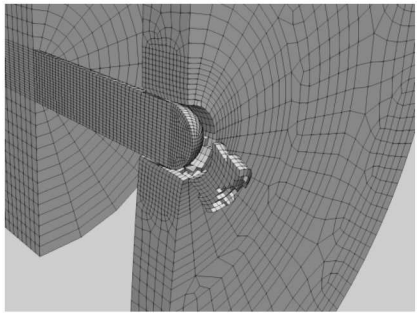


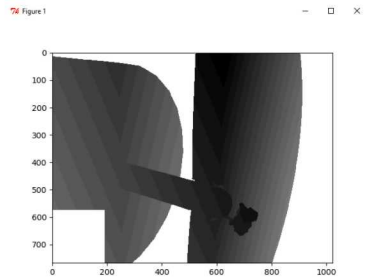
Dimension 100 Results

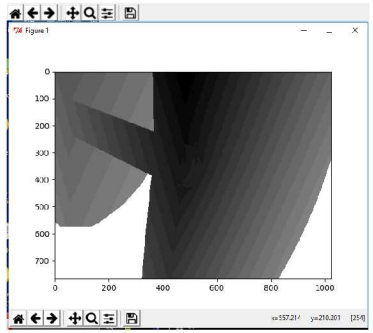


Cinema Floating Point vs RGB

- Comparison of VideoSwarm from Von Mises values vs. RGB
- Difficulty getting data (consequently, ensemble size is small)
  - Couldn't get Catalyst to generate Cinema outputs
  - Manually created 10 run ensemble using ParaView 5.5.2 and 'export scene' with Cinema output (same view in all, single static camera)
    - RGB (as jpg)
    - composable + recolorable
      - Von Mises, depth buffer (as Z)
      - Luminance (as png)

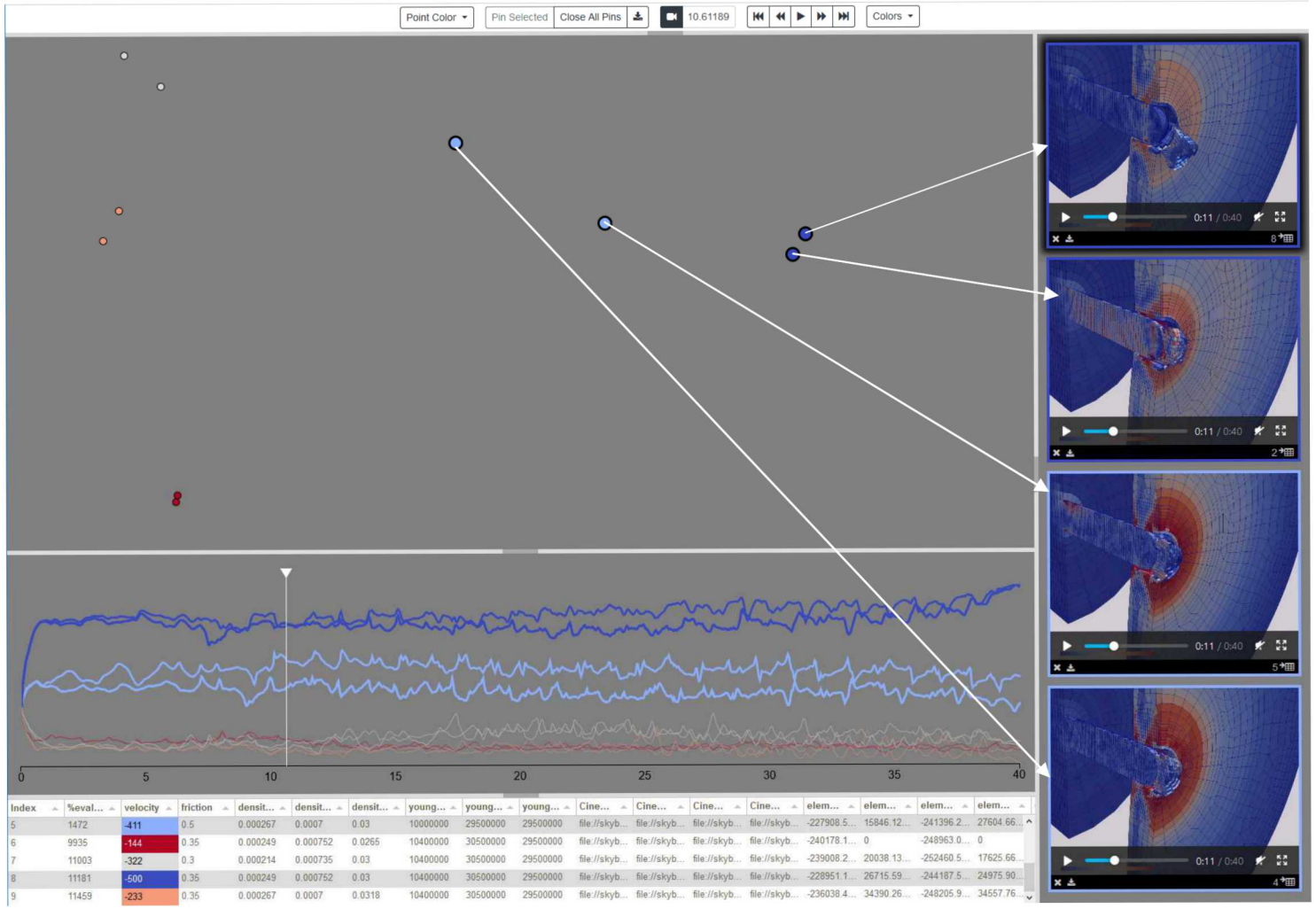






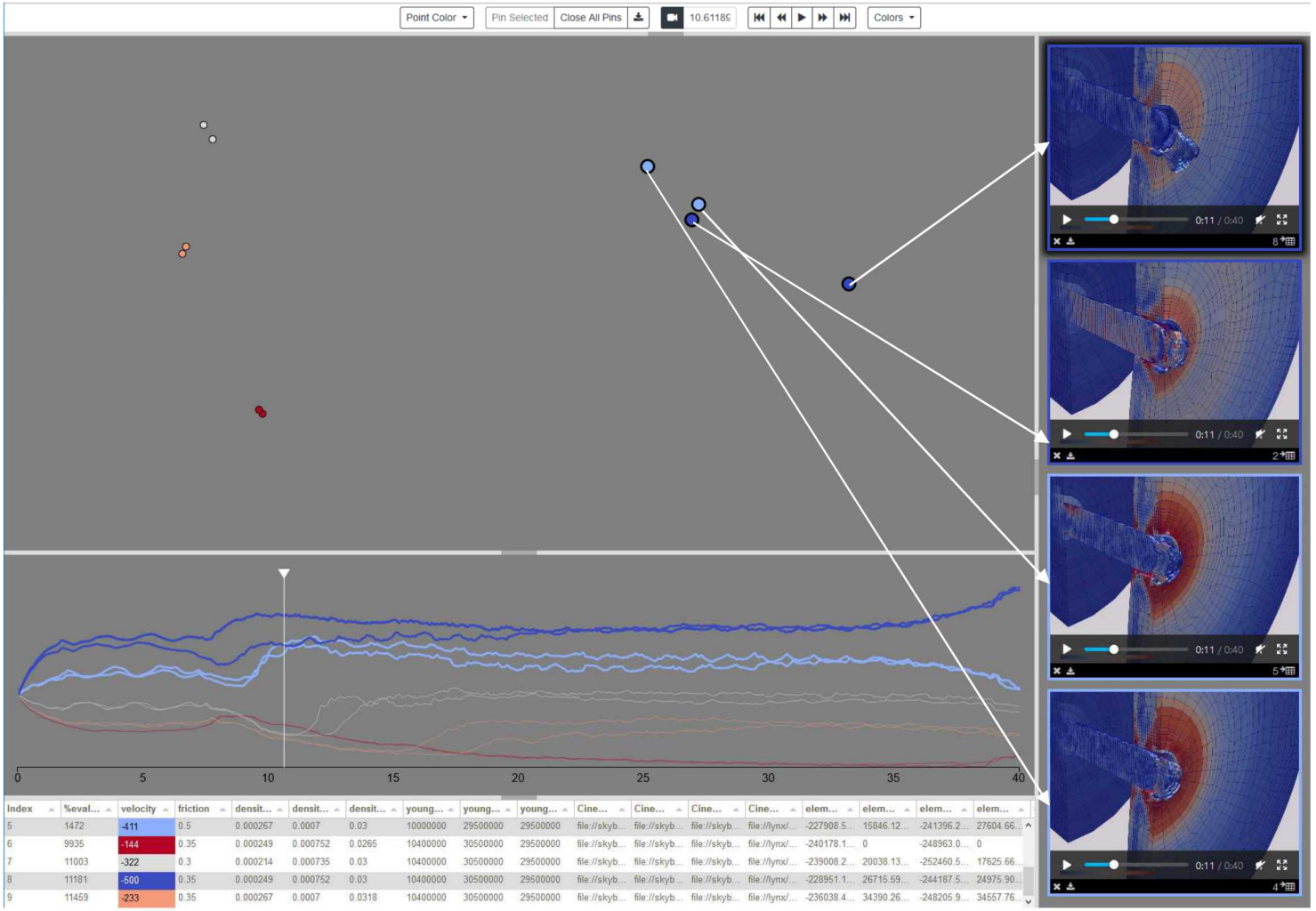
- Data definition
  - Floating point values/pixel where object present
  - Depth buffer determines object vs. background pixels
    - Depth buffer bugs (reduced by shifting view)
    - Problem comparing undefined (background) pixels
    - Von Mises values are positive, so set undefined = 0

Cinema Von Mises Floating Point



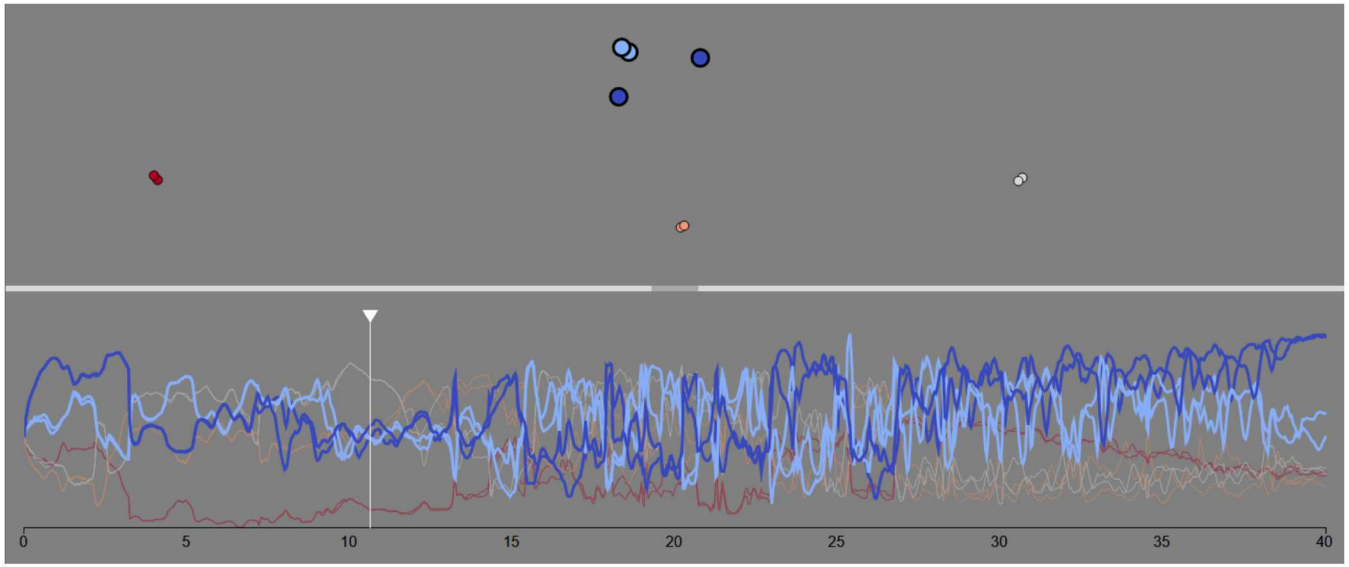
Index	%eval...	velocity	friction	densit...	densit...	densit...	young...	young...	young...	Cine...	Cine...	Cine...	Cine...	elem...	elem...	elem...	elem...
5	1472	-411	0.5	0.000267	0.0007	0.03	10000000	29500000	29500000	file://skyb...	file://skyb...	file://skyb...	file://skyb...	-227908.5...	15846.12...	-241396.2...	27604.66...
6	9935	-144	0.35	0.000249	0.000752	0.0265	10400000	30500000	29500000	file://skyb...	file://skyb...	file://skyb...	file://skyb...	-240178.1...	0	-248963.0...	0
7	11003	-322	0.3	0.000214	0.000735	0.03	10400000	30500000	29500000	file://skyb...	file://skyb...	file://skyb...	file://skyb...	-239008.2...	20038.13...	-252460.5...	17625.66...
8	11181	-500	0.35	0.000249	0.000752	0.03	10400000	30500000	29500000	file://skyb...	file://skyb...	file://skyb...	file://skyb...	-228951.1...	26715.59...	-244187.5...	24975.90...
9	11459	-233	0.35	0.000267	0.0007	0.0318	10400000	30500000	29500000	file://skyb...	file://skyb...	file://skyb...	file://skyb...	-236038.4...	34390.26...	-248205.9...	34557.76...

Cinema RGB

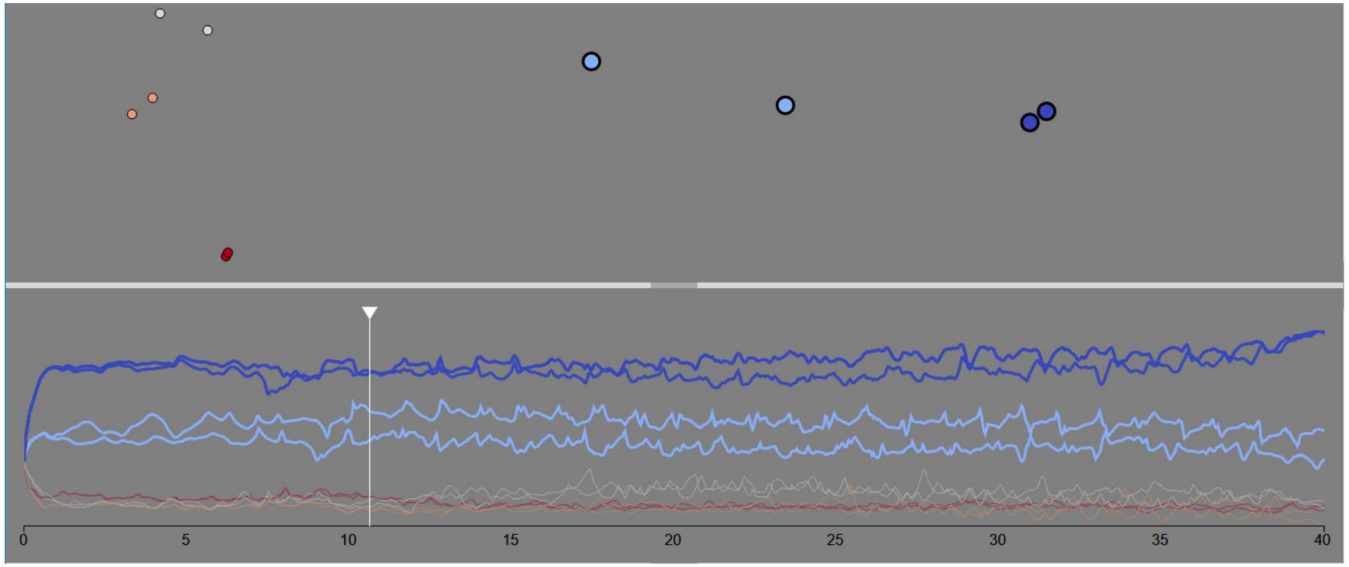


Index	%eval...	velocity	friction	densit...	densit...	densit...	young...	young...	young...	Cine...	Cine...	Cine...	Cine...	elem...	elem...	elem...	elem...
5	1472	-411	0.5	0.000267	0.0007	0.03	10000000	29500000	29500000	file://skyb...	file://skyb...	file://skyb...	file://lynx/...	-227908.5...	15846.12...	-241396.2...	27604.66...
6	9935	-144	0.35	0.000249	0.000752	0.0265	10400000	30500000	29500000	file://skyb...	file://skyb...	file://skyb...	file://lynx/...	-240178.1...	0	-248963.0...	0
7	11003	-322	0.3	0.000214	0.000735	0.03	10400000	30500000	29500000	file://skyb...	file://skyb...	file://skyb...	file://lynx/...	-239008.2...	20038.13...	-252460.5...	17625.66...
8	11181	-500	0.35	0.000249	0.000752	0.03	10400000	30500000	29500000	file://skyb...	file://skyb...	file://skyb...	file://lynx/...	-228951.1...	26715.59...	-244187.5...	24975.90...
9	11459	-233	0.35	0.000267	0.0007	0.0318	10400000	30500000	29500000	file://skyb...	file://skyb...	file://skyb...	file://lynx/...	-236038.4...	34390.26...	-248205.9...	34557.76...

Cinema Von Mises Dim 2 vs 10

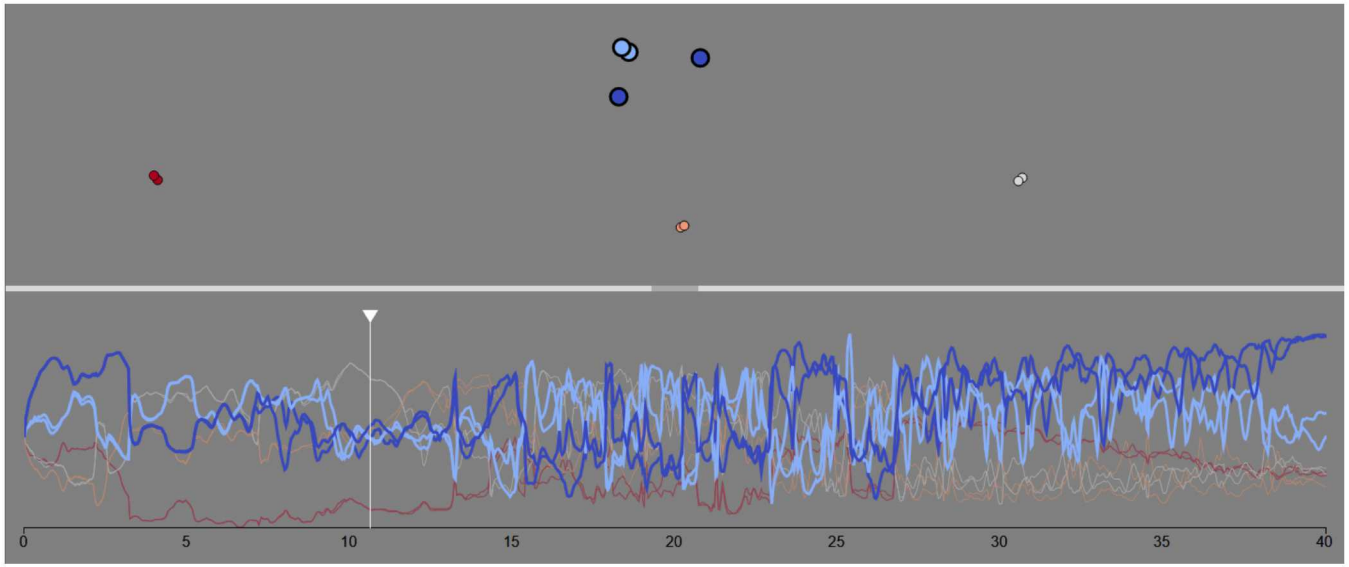


Dimension 2

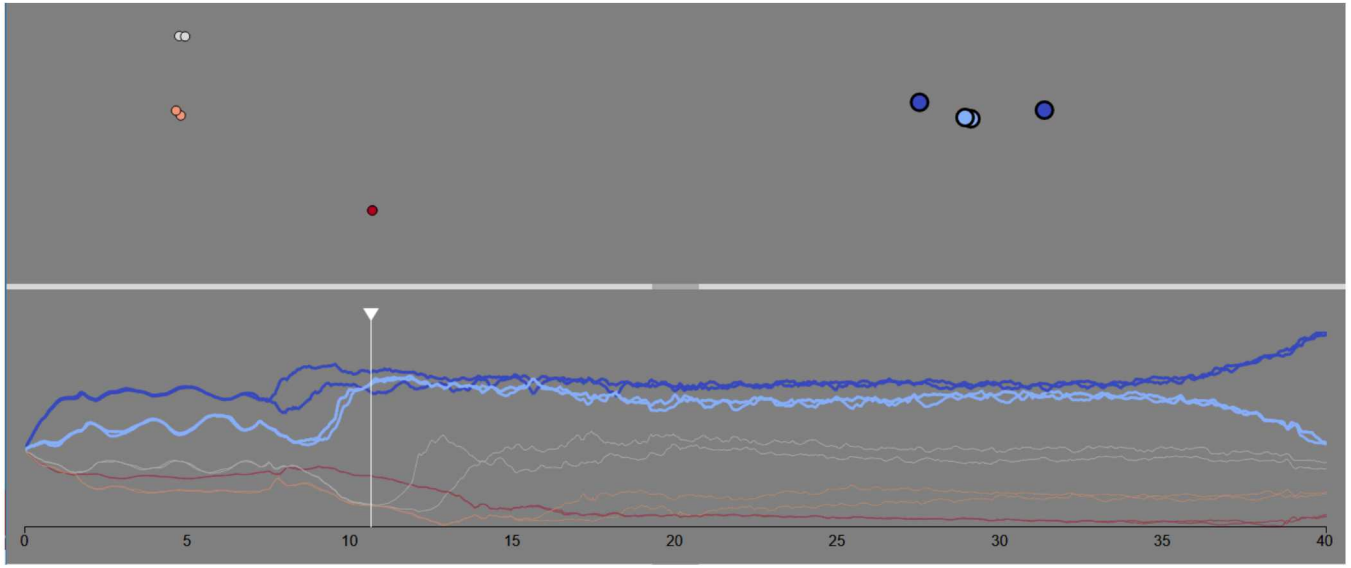


Dimension 10

Von Mises Dim 2 vs RBG Dim 2



Von Mises
Dimension 2



RGB
Dimension 2

Conclusions

Sandia National Laboratories

- Temporal MDS improves continuity issues
- Dimension 10 is good default choice
- Floating point provides no advantage*
  - Comparison of undefined pixels problematic
    - NANs for undefined pixels
    - Variables with range crossing zero cannot set undefined pixels to zero
    - Potential for no intersection between defined pixels (fragments)
  - Eigenvector rank flipping problem greater than with RGB
  - Cinema generation issues, while RGB outputs universally generated

*Caveats

- Cinema ensemble size too small
- Depth buffer bug impacts comparison

Multidimensional Scaling

To assign each video a 2D coordinate representation, we use a dimension reduction technique called Multi-Dimensional Scaling (MDS).

- Time trajectories are plotted as the first coordinate of MDS versus time for each video.
- First compute pair-wise distances between each frame of each video in the ensemble.

$$D_t = \begin{bmatrix} d(\mathbf{f}_{1t}, \mathbf{f}_{1t}) & d(\mathbf{f}_{1t}, \mathbf{f}_{2t}) & \dots \\ d(\mathbf{f}_{2t}, \mathbf{f}_{1t}) & d(\mathbf{f}_{2t}, \mathbf{f}_{2t}) & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}$$

$$d(\mathbf{f}_{it}, \mathbf{f}_{jt}) = \sqrt{\sum_k (f_{itk} - f_{jtk})^2}$$

Multidimensional Scaling

Second, “double-center” the distance matrix.

$$B = -\frac{1}{2} H D^2 H$$

$$H = I - 11^T / n$$

Multidimensional Scaling

Third, compute the eigenvalue decomposition of B, keeping the two largest positive eigenvalues λ_1, λ_2 and corresponding eigenvectors $\boldsymbol{e_1}, \boldsymbol{e_2}$.

$$B = E\Lambda E^T$$

- The MDS coordinates are given by the first two columns of $E\Lambda^{1/2}$.
- (In fact, keep *k* dimensions of each reduction for the purpose of aligning frames, described in the next slide.)

Multidimensional Scaling

Finally, align the MDS visualizations per frame using an optimal orthogonal transformation.

- If P and Q are matrices containing the MDS coordinates of two consecutive time steps, we compute (using the Singular Value Decomposition)

$$A = P^T Q \qquad R = V_k U_k^T$$

$$A = U \Sigma V^T$$

- R gives an optimal transformation (rotation and/or reflection) matrix which best aligns P and Q using up to *k* dimensions.
- The frames are aligned sequentially from the last frame to the first frame using *k* dimensions, then projected into 2D for the final visualization.