Proposal: Visual Exploration of Multi-faceted Halo Data Sets

Xiaotong Liu
Meraj Khan

1. Introduction

Cosmological simulations follow the formation of nonlinear structure in dark and luminous matter. In the Dark Sky Simulations project, each simulation is typically evolved over 14 billion years from very small fluctuations in an otherwise uniform distribution, using gravitational N-body integrators in an expanding background Universe. The particles in cosmological simulations cluster into gravitationally bound structures that pull in baryonic matter that form stars, galaxies, and clusters of galaxies.

One fundamental research topic in cosmological simulation is the visualization of halos that are formed by particles and defined by mass, position, and velocity and a set of additional properties of the halo, such as angular momentum, characteristic radius, spin parameter. These multi-faceted attributes complicate the exploration of halos in such high dimensional data space, as it requires investigating the multiple dimensions and their specific values to understand the multi-faceted properties of the halos.

Our goal of this work is to design a visualization framework that assists users to analyze the multi-faceted properties of the halos for visualizing halos of interest in cosmological simulations. A multi-faceted view of halo properties can reveal the relationships between the facets and prompt selecting of interest. We propose to combine multi-dimensional visualization techniques and particle visualization to support visual exploration of multi-dimensional halo data sets. We expect to provide linked views of the halos as well as their multi-faceted properties. Various interactions such as zooming and filtering will also be supported to allow users to explore the subspace of the data sets.

2. Data set

In this work, we use the IEEE VIS contest data set in 2015, which consists of three primary data types: the raw particle data, the Halo catalog data, and the merger tree data.

First, the raw particle data contains a position vector, velocity vector, and unique particle identifier. There are approximately 100 time steps, each stored in the SDF. This format is composed of a human readable ASCII header followed by raw binary data.

Second, the Halo Catalog data groups sets of gravitationally bound particles together into coherent structures. It has the information about a given halo's position, shape, and size, are a number of statistics derived from the particle distribution, such as angular momentum, relative concentration of the particles, and many more. These catalogs are stored in both ASCII and binary formats.

Third, the Merger Tree data links the individual halo catalogs that each represents a snapshot in time, thereby creating database. Each merger tree forms a sparse graph that can then be analyzed to use quantities such as halo mass accretion and merger history to inform how galaxies form and evolve through cosmic time. Merger tree data are also distributed in both ASCII and BINARY formats.

3. Tasks

The main task to address in this work is to visualize the multi-faceted halo properties for exploring and comparing halos in cosmological simulations. The particles that compose a halo, which are defined by mass, position, and velocity, are used to derive additional properties of the halo such as angular momentum, characteristic radius, and spin parameter. Given many halo properties, users may have varying interest in exploring halos of varying property combinations. To navigate through such rich data, one can rely on search technologies to find relevant information. Search tools typically return a ranked list of entities whose content is highly related to a set of user-supplied range queries. However, without sufficient prior knowledge, the query specification can be difficult as the enormous multidimensional space complicates the search of potentially interesting halos. This high-dimensional property space poses a challenge to analysis tasks where users try to understand and compare different halos that are described by multiple properties. To enable easy query specification for selecting halos of interest, users need to examine the value distributions of the halo properties as well as their relationships. Therefore, visualizations that aid users to view and compare halo properties and the relationships of the properties are of great importance.

Furthermore, the latency of interaction, namely, how much time it takes for the system to respond to the user action, matters for a smooth exploration process. Due to the large number of halos, responsive and flexible interaction schemes are required to allow users to zoom into the subspace of data sets with acceptable latency.

4. Software

Yt.py is used to read and process the raw particle data in SDF format. Yt.py is a python package for analyzing and visualizing volumetric, multi-resolution data from astrophysical simulations, radio telescopes, and a burgeoning interdisciplinary community. It can seamlessly handle simulation output files to enable effective analysis.

D3.js is used to create the PCP view, and support interactions such as brushing and linking. D3.js is a JavaScript library for manipulating documents based on data, which has various visualization components and a data-driven approach to DOM manipulation on Web.

WebGL is used to render the halos in the Halo view, and support interactions changing camera position and orientation of the view, and controlling the color and opacity of the halos. WebGL (Web Graphics Library) is a JavaScript API for rendering interactive 3D computer graphics and 2D graphics, which can be mixed with other HTML elements and composited with other parts of the page or page background.

Node.js is used to build a server for querying data on demand. Node.js is a platform built on Chrome's JavaScript runtime for easily building fast, scalable network applications. It uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.

We will create a web-based system that assists users to explore multi-faceted halo data sets.

5. Visualization Design

We propose to combine multi-dimensional visualization techniques and particle visualization to support visual exploration of multi-dimensional halo data sets. To this end, we employ parallel coordinates plot (PCP) to visualize the multi-faceted halo properties in the *PCP view*. PCP is a common way of visualizing high-dimensional data, representing a point in multiple-dimensional space as a polyline with vertices on parallel axes, where the position of the vertex on an axis corresponds to the value of that dimension. In our design, each axis corresponds to one property of the halo, and the point at which the polyline intersects represents the value of the point in that property. An example of the PCP view is illustrated in Figure 1. One challenge in designing the PCP view is to enable responsive interaction via brushing and linking, as a conventional PCP often gets less responsive with thousands of polylines. To solve this, we will resort to *progressive rendering*. This will be done by arranging all the multi-dimensional data points in a queue, rendering a subset of the data points per frame, and progressively building up the final rendering result.

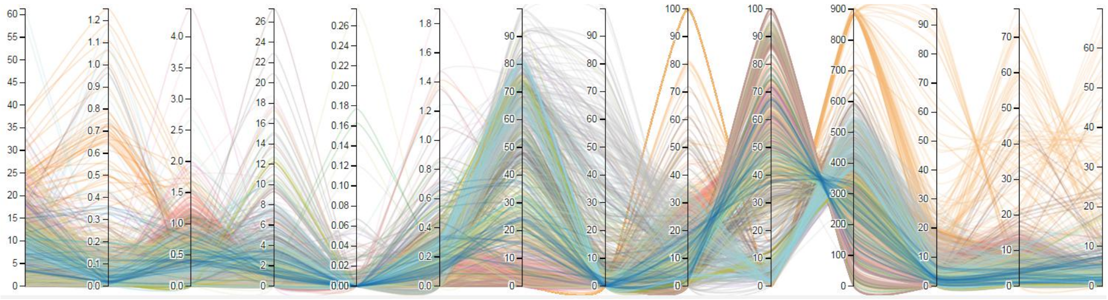
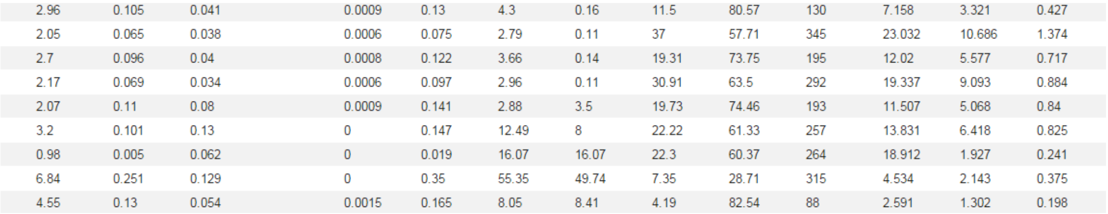


Figure 1: an illustration of the PCP view.

This PCP view will be linked with two additional views: the *Property view* and the *Halo view*. The Property view (Figure 2) shows the exact values of the properties in an interactive table that allows users to select one row to examine the numerical properties of the particular halo, while the corresponding polyline is highlighted in the PCP view. On the other hands, brushing the PCP axes will also highlight the selected polylines in the Property view.



2.96	0.105	0.041		0.0009	0.13	4.3	0.16	11.5	80.57	130	7.158	3.321	0.427	
2.05	0.065	0.038		0.0006	0.075	2.79	0.11	37	57.71	345	23.032	10.686	1.374	
2.7	0.096	0.04		0.0008	0.122	3.66	0.14	19.31	73.75	195	12.02	5.577	0.717	
2.17	0.069	0.034		0.0006	0.097	2.96	0.11	30.91	63.5	292	19.337	9.093	0.884	
2.07	0.11	0.08		0.0009	0.141	2.88	3.5	19.73	74.46	193	11.507	5.068	0.84	
3.2	0.101	0.13		0	0.147	12.49	8	22.22	61.33	257	13.831	6.418	0.825	
0.98	0.005	0.062		0	0.019	16.07	16.07	22.3	60.37	264	18.912	1.927	0.241	
6.84	0.251	0.129		0	0.35	55.35	49.74	7.35	28.71	315	4.534	2.143	0.375	
4.55	0.13	0.054		0.0015	0.165	8.05	8.41	4.19	82.54	88	2.591	1.302	0.198	

Figure 2: an illustration of the Property view.

Once users select the halos of interest in the PCP view and the Property view, the *Halo view* (Figure 3) provides users with the most direct and intuitive view of the halo structures using particle rendering. For a certain halo, the particles that fall into the radius of the halo will be shown.

These three views will be linked together to support the exploration process: users can brush the PCP view and manipulate the property view to select halos of interest in a subspace, and examine the corresponding halo structures in the halo view.

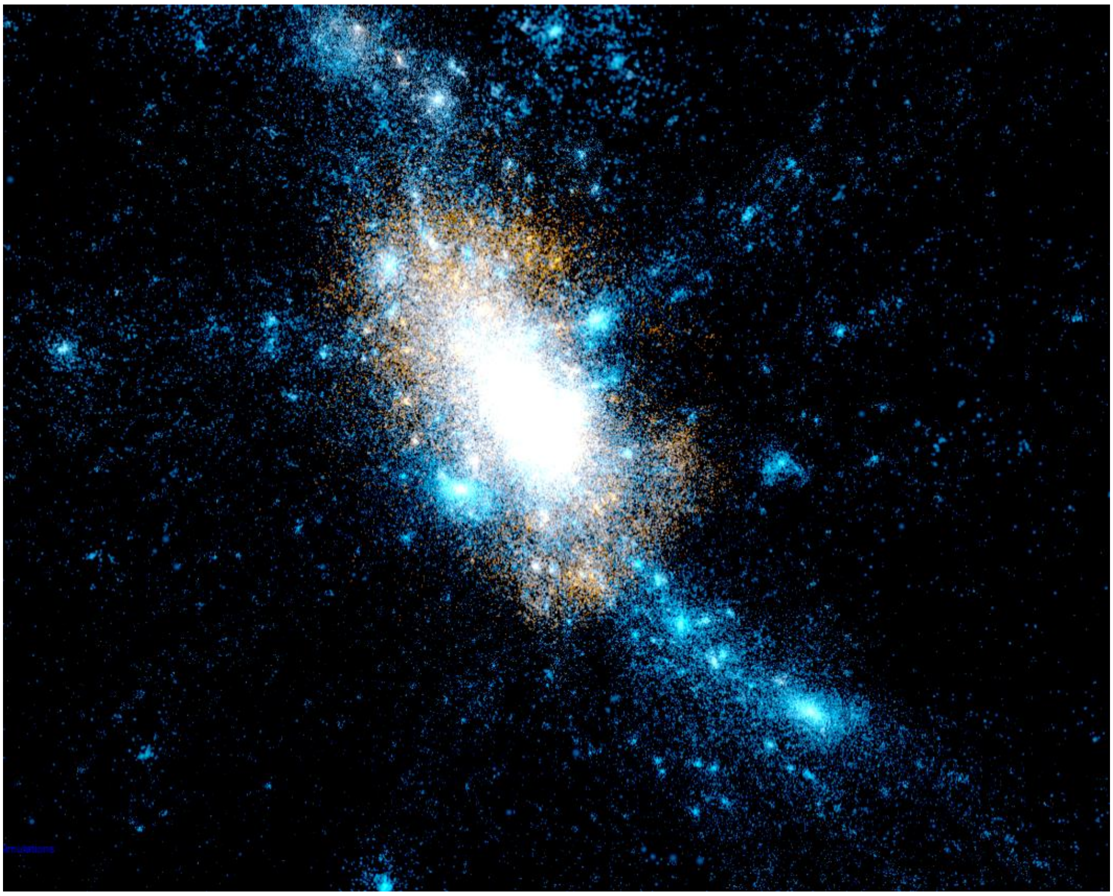


Figure 3: an illustration of the Halo view.

6. Expected Results and Evaluations

We have downloaded the entire IEEE VIS contest data in 2015, and process the raw particle data using yt.py. We have also implemented the PCP view using D3.js. We expect to implement the linked views of the halos as well as their multi-faceted properties following our design choices in Section 5. Various interactions such as zooming and filtering will also be supported to allow users to explore the subspace of the data sets. To demonstrate the effectiveness and usefulness of our approach, we will perform a case study on the IEEE VIS contest data set in 2015 using our visualization system.