Contents lists available at ScienceDirect

Visual Informatics

journal homepage: www.elsevier.com/locate/visinf

Research article

Time analysis of regional structure of large-scale particle using an interactive visual system

Yihan Zhang [a,b], Guan Li [a], Guihua Shan [a,b,*]

[a] Computer Network Information Center, Chinese Academy of Sciences, China
[b] University of Chinese Academy of Sciences, China

ARTICLE INFO

Article history:
Received 1 September 2021
Received in revised form 28 February 2022
Accepted 23 March 2022
Available online 30 March 2022

Keywords:
Visual analytics
Frame interpolation
Interactive
Particle region structure

ABSTRACT

N-body numerical simulation is an important tool in astronomy. Scientists used this method to simulate the formation of structure of the universe, which is key to understanding how the universe formed. As research on this subject further develops, astronomers require a more precise method that enables expansion of the simulation and an increase in the number of simulation particles. However, retaining all temporal information is infeasible due to a lack of computer storage. In the circumstances, astronomers reserve temporal data at intervals, merging rough and baffling animations of universal evolution. In this study, we propose a deep-learning-assisted interpolation application to analyze the structure formation of the universe. First, we evaluate the feasibility of applying interpolation to generate an animation of the universal evolution through an experiment. Then, we demonstrate the superiority of deep convolutional neural network (DCNN) method by comparing its quality and performance with the actual results together with the results generated by other popular interpolation algorithms. In addition, we present PRSVis, an interactive visual analytics system that supports global volume rendering, local area magnification, and temporal animation generation. PRSVis allows users to visualize a global volume rendering, interactively select one cubic region from the rendering and intelligently produce a time-series animation of the high-resolution region using the deep-learning-assisted method. In summary, we propose an interactive visual system, integrated with the DCNN interpolation method that is validated through experiments, to help scientists easily understand the evolution of the particle region structure.

1. Introduction

In astronomy, scientists apply hardware equipment such as observational astronomical facilities to obtain cosmic data. Then, they analyze these data and use them as a reference to publish new findings. However, the current level of hardware devices restricts the observable range and precision. Most of the universe is mysterious to humans. For instance, despite dark energy being the most dominant element of cosmological energy density (Somerville and Davé, 2015), we know nearly nothing about it. To learn more about the universe, the numerical simulation method was proposed, which plays a critical role in cosmology. The simulation subdivides the universe into particles, determines the force on every particle by numerically solving Poisson's equation within a comoving frame, and evolves such a system forward in discrete time-steps (Somerville and Davé, 2015). N-body simulation methods include particle-based, mesh-based, and hybrid types. A hierarchical O (N log N) algorithm presented by Barnes is a typical code (Barnes and Hut, 1986). At present, popular methods combine the advantages of particle-based method (tree code (Barnes and Hut, 1986)) and particle–mesh method, to obtain more accurate results in a small range and faster computation in a large range (e.g.GADGET-2 Springel, 2005). With the assistance of numerical simulation, scientists adjust the scope and precision of exploration by inputting some initial parameters into the supercomputer to simulate the evolution of the universe. This approach assists scientists in verifying presented hypotheses, improving theoretical models, and understanding the evolution of the universe.

By conducting numerical simulation, astronomers analyze the cosmic structure built upon dark matter represented by particles. As the research advances, specialists demand improved simulation, not only in terms of the scope but also the number of particles. These requirements pose new challenges for the storage and calculation of supercomputers. Domain experts now store temporal data at fixed intervals, explore meaningful properties,

* Corresponding author at: Computer Network Information Center, Chinese Academy of Sciences, China.
E-mail addresses: zhangyihan@cnic.cn (Y. Zhang), liguan@sccas.cn (G. Li), sgh@sccas.cn (G. Shan).

https://doi.org/10.1016/j.visinf.2022.03.004

and observe feature trends. The analysis works better with a smooth animation. One concern is how to gain such an animation with high interactivity. With no other operation, frames assemble simply at wide intervals, which results in a confusing video. Even though storage remains those necessary temporal data, researchers take the speed of computation into account. The problem is two-sided in that it has a) limited storage capability and b) slow computation speed.

Producing a smooth video given limited data is a popular theme in video post-processing. In general, scientists interpolate frames in a video by using an interpolation algorithm to increase the frames per second (fps). Interpolation generates intermediate frames between two given images in a video sequence (Werlberger et al., 2011). The interpolated frame supplies a transition between the sequences, making the variation more natural. Moreover, this method may restore the appearance of characteristics that vanished in the interval. Conventional interpolation algorithms calculate the motion of image intensity, such as the optical flow method. In addition, a few scientists consider the change of every pixel in time, as in the phase-based method. Recently, some researchers have combined DCNN with traditional interpolation methods that have high accuracy. These methods provide ideas on whether astronomers can apply interpolation in the evolution of the universe, whether the accuracy and speed of interpolation are acceptable, and whether the selected methods can be improved.

To verify these ideas, we designed an experiment in which we compared four different interpolation methods applied to the evolution of the universe. As the particle is the main element of simulation, we first confirmed two particle-based algorithms. The simulation environment is a 3D space, which facilitates image rendering and the calculation of the position of every particle at the interpolation time. One method is to directly calculate the particle position according to the interpolation number. Another is based on variable velocity and simulation parameters. Then, we picked a phase-based method, which is one of the most excellent interpolation algorithms for images. An approach that we used is Super SloMo (Jiang et al., 2018), an end-to-end DCNN based on U-Net (Ronneberger et al., 2015). After training the network using astronomic datasets, we designed an evaluation experiment, through which we proved the feasibility of applying these interpolation approaches in astronomy. In addition, we compare the results of computation performance and image authenticity. To improve the robustness of interpolation, we also input some unseen cases. Finally, we pick the most balanced method, Super SloMo, to integrate with our interactive visual system, PRSVis.

PRSVis involves rendering, selection, and time-series analysis. At first, the system shows a 3D volume rendering, a visualization of the particle position. Users move the mouse to rotate and draw a concerned area. The interface then displays the local high-resolution rendering. After a click on the generation button, the system generates a smooth time-series animation of the selected region by the Super SloMo.

In summary, this study explores the feasibility of applying the interpolation approach in the analysis of the large-scale structure of the universe. We evaluate three different types of interpolation methods in terms of speed and storage. To come up with an integrated process, we present an interactive visual analytics system that allows users to begin with global rendering, confirm local selection, and receive smooth temporal animation. Specifically, the primary contributions of our study are as follows:

- designing an experiment to prove the feasibility of adopting the interpolation approach in cosmic structure analysis,
- comparing four different interpolation algorithms to confirm the highly interactive interpolation approach, and
- developing an integrated interactive analytics system to improve the understanding of the particle region structure in temporal terms.

2. Related work

Snapshots of simulation combine interpolation algorithms to present a smooth evolution animation. In this section, we provide a research overview of the evolution of the universe. Then, we present a detailed introduction and discuss the related work on the interpolation algorithms.

2.1. Evolution of the universe

In astronomy, the universe represents a specific research object. Scientists explore how the universe began, and how it evolves and develops. G. Gamow (1948) (Gamow, 1948) described the universe as being in a state of uniform expansion through the discovery of the redshift in the spectra of distant galaxies; he also inferred the age of the universe according to natural radioactive isotopes. Novikov (1983) projected the physical processes of the evolution of the universe, including violent reactions between elementary particles (Novikov, 1983). In addition, numerous studies have attempted to explain the evolution of the universe, for example, homogeneous and isotropic universe models (Einstein, 1919) (Einstein, 1919), big bang theory (Lemaître, 1927) (Lemaître, 1927), and a circular model (Paul, 2002) (Steinhardt and Turok, 2002). However, astronomy is a domain that one can see but not touch (Ratra and Vogeley, 2008). Most of the aforementioned findings began with hypotheses and only provided possible models. According to imaging data obtained by the Hubble Space Telescope, GN-z11, the most distant galaxy to the date (Jiang et al., 2021), is approximately 32 giga light-years (Gly) away from Earth, while the observable universe is approximately 46 Gly (Halpern and Tomasello, 2016). In this situation, researchers have conducted extensive work on the evolution of the universe using N-body numerical simulation. Scientists simulate the universe far beyond the observable scale of current hardware technology.

Astronomers transform structural formation models into an initial value problem by using numerical simulation (Bertschinger, 1998). In other words, they adjust the input parameters of simulations to obtain different results and verify their hypotheses about universe evolution. For instance, Davis et al. (1985) (Davis et al., 1985) simulated the large-scale structure of the universe several times by parameter change. Through their simulation results, they concluded that galaxies formed at high peaks of the initial density field instead of randomly. Scientists have adopted many approaches to increase the precision of simulation, for example, by applying resolution change methods. The grid-based N-body algorithms first enable dark matter simulation with more than 10^5 particles (Bertschinger, 1998). Through Moore's law combined with heavy infrastructure investments, scientists now could conduct simulations with over half a trillion particles. Although current simulations have such high precision, Kuhlen (Kuhlen et al., 2012) pointed out that one of the greatest needs for the dark energy (DE) problem is a tremendous increase in the number of high resolution. Together, the improvement room for dark matter (DM) problem is the internal structure of subhalos, which also need further performance.

Some scientists conduct in-situ visualization. For example, Ahrens et al. (2014) (Ahrens et al., 2014) presented an image-based in-situ method that defines a set of operations before simulation, which the supercomputers calculate and analyze together. This method requires the user to be familiar with simulation data to have enough a priori knowledge and certain goals. The 4D street view method (Kageyama and Sakamoto, 2020) scatters multiple omnidirectional cameras to record the simulation and then explores a collection of images. This method extracts images along a path in space, such as by human movement, but the images become distorted in preserving the 4D view, which makes them unsuitable for observing the evolution of the universe. Yamaoka et al. (2019) presented a mechanism to adaptively control the timestep sampling intervals, which is calculated by the kernel density estimation. This statistical method highly relies on a user-defined parameter that is unstable. In our study, we focus on the evolution of the local region that is interactively selected. We choose interpolation to generate animations of sub-structures after magnifying this region.

2.2. Interpolation algorithm

Generating intermediate frames between two given images in a video sequence is an essential task (Werlberger et al., 2011) in video post-processing. Before the emergence of artificial intelligence, scientists roughly divided the traditional interpolation methods into two categories: Lagrange and Euler methods (Meyer et al., 2015).

The Lagrange method considers the motion of pixels in space, for example, the optical flow method. The optical flow is the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between the observer and the scene (MARKMAN, 1979; Warren and Strelow, 1985). Finding out the pattern of object motion is the core of the optical flow method, which has made progress over the years. Lucas and Kanade (1981) presented an image registration technology that takes advantage of the spatial intensity gradient of images (Lucas, 1981). They considered the image information, which finds the best match between two images more quickly than the fixed positions. The optical flow algorithm is remarkable, with a number of researchers further developing it. Approaches based on this algorithm are classified into two types: continuous and discrete (Baker et al., 2011). The continuous optimization contains gradient descent algorithms (see Baker and Matthews (2004) for details), extremal methods, and variational approaches (Black and Jepson, 1996; Black and Anandan, 1996). In addition, research is optimized by decoupling the data (Trobin et al., 2008; Wedel et al., 2008) together with the application of linear programming (Seitz and Baker, 2009). For discrete optimization, approximations are always used to transform data from discrete to continuity. Boykov (2001) applied the graph cuts (Boykov et al., 2001), and Sun (2003) used the belief propagation (Sun et al., 2003). Lempitsky (2008) invented fusion flow to refine the result by local replacement (Lempitsky et al., 2008), and Darabi combined the state space definition with flow estimation (Black and Anandan, 1991; Glocker et al., 2008; Lei and Yang, 2009). In summary, all these methods deal with images in space.

Some scientists improved the quality of interpolation frames from other aspects. Sadek et al. (2012) cut the temporal and spatial area in time coherent videos to handle occlusions (Sadek et al., 2012). Qingchun Lu et al. (2016) presented an occlusion handling algorithm for motion-compensated frame interpolation, which considers not only the unidirectional motion vector fields but also occlusion conditions (Lu et al., 2016).

By contrast, the Euler method takes into account the color change of pixels in time. This method computes weights, which can magnify the subtle motions. Abe et al. (2014) averaged the pixels in input videos, gaining a trace thousands of times than original (Davis et al., 2014). Meal et al. (2013) highlighted and eliminated the change in pixels by complex-valued image pyramids, a phase-based approach (Wadhwa et al., 2013). Soheil et al. (2012) demonstrated a patch-based optimal method with a natural transition between two images, which prevents generating ghost artifacts (Darabi et al., 2012). In general, the Euler method performs excellently on tiny objects.

Machine learning and deep learning have developed rapidly in recent years. Numerous results have combined these techniques with traditional interpolation methods. For example, Niklaus et al. (2017) presented a DCNN that regards the interpolation frames as input frames (Niklaus et al., 2017). This approach estimates a spatially adaptive convolution kernel for each pixel instead of motion estimation, which is more direct. Liu et al. (2017) performed deep voxel flow (DVF) for synthesizing video frames (Liu et al., 2017). The combinations perform well and motivate us to apply such methods in astronomy.

3. Background

We evaluated four interpolation methods in our experiments, which contain two different particle-based algorithms, an improved phase-based method (Meyer et al., 2015), and Super SloMo, a DCNN method (Jiang et al., 2018). In Section 3.1, we describe the theories behind these approaches and the reason for choosing them. The Super SloMo, which we finally adopted, is presented in detail in Section 4.2. Furthermore, to validate the evaluation results, we introduce two test methods in our comparison tests: peak signal-to-noise ratio (PSNR) and structural similarity (SSIM).

3.1. Comparative methods

The particle-based algorithms take the character of dark matter simulation into account. In the cosmic N-body simulation, the fundamental unit is the dark matter particle, which is distributed in the 3-D simulation space. Together, the time-series animation that we conduct shows the universe structure formation through particle position visualization. We regard the particle motion in space as a position shift in animation presentation. Given that every particle moves linearly in a flash, we calculate the particle position at all times through two adjacent timestep files. In other words, when we design experiments, we should guarantee that two timestep files are at small intervals. We gain the interpolation frames by calculating the particle position at the interpolated time. For the first method, we assume that the velocity of particles is the same during interpolation intervals. The position calculation formula of every particle is expressed as

$$p(x,y,z)_{frame(i)} = p(x_{t_1}, y_{t_1}, z_{t_1}) + \frac{i}{n+1} * ((x_{t_2} - x_{t_1}), (y_{t_2} - y_{t_1}), (z_{t_2} - z_{t_1})) \tag{1}$$

where p is the particle position, t_1 and t_2 respectively represent the timesteps of two interpolation frames, n is the total interpolation frame number between two initial frames and i is the sequence number of interpolation frames ordered by time.

The second particle-based method is widely used in astronomy, and assumes that the accelerated velocity varies with time during the interpolation period. At the same time, this method adds some initial simulation parameters into the calculation, such as Hubble, Omega0 and OmegaLambda. The formula is expressed as

$$p(t) = p_0 + \int_{t_0}^{t} [v_0 + c(t - t_0) + 3f(t - t_0)^2] dt, \tag{2}$$

where $p(t)$ indicates the particle position at time t, p_0, v_0, t_0 respectively represent the particle initial position, velocity, and time. The function f is related to acceleration, and c is the hyper parameter related to the simulation.

The improved phase-based algorithm is classified as a type of Euler method. The phase-based algorithm is based on the cognition that pictures can be regarded as a superposition of 2D discrete sinusoidal signals. The basic structure of this method includes the initial image signal isolation, every sinusoidal signal phase calculation, and finally phase difference processing, which

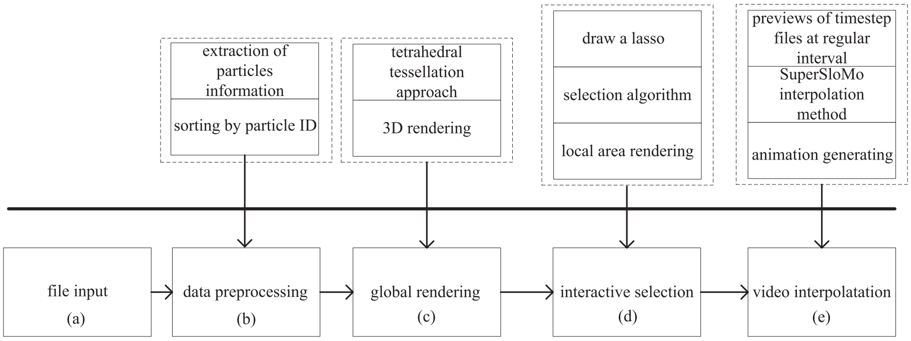

extraction of particles information
sorting by particle ID
tetrahedral tessellation approach
3D rendering
draw a lasso
selection algorithm
local area rendering
previews of timestep files at regular interval
SuperSloMo interpolation method
animation generating
file input (a)
data preprocessing (b)
global rendering (c)
interactive selection (d)
video interpolatation (e)

Fig. 1. The analysis framework consists of five main steps: (a) This step scores output simulation files at every timestep. (b) The data preprocessing refines the files in (a) and rearranges the particle information via particle ID. (c) After selecting the initial timestep file, we render all particles in the simulation using the tetrahedral tessellation approach. (d) An area is interactively selected. Using our selection algorithm, we confirm the final region as the leading part in the time-series animation. (e) Video interpolation generates the local structure animation using the DCNN method, Super SloMo.

is used to generate in-between frames. In the application of this theory, confirming the actual phase shift for sinusoidal function is challenging. Meyer et al. (2015) (Meyer et al., 2015) proposed a boundary phase-based method that corrects phase shifts and breaks the limitation of the pixel motion range using the multi-scale pyramid. In addition, the phase-based method employs a big constant to restrict the phase motion. After calculating the sinusoidal phase difference, the amplitude and frequency are also needed. This approach linearly blends the amplitude of initial images and linearly mixes the remaining frequency through the low-pass residual. Finally, this method obtains advanced levels among traditional interpolation algorithms. This improved phase-based algorithm considers plenty of details and provides excellent performance, leading us to choose it as a comparative method in our experiments.

Super SloMo is the emphasis method that we finally adopted. This approach combines optical flow method and DCNN, which performs better than DVF by evaluating several mainstream ways in the datasets of Adobe240-fps (Su et al., 2017), UCF101 (Soomro et al., 2012), SlowFlow (Janai et al., 2017) and Sintel (Janai et al., 2017). However, this method has not been applied in astronomy. We explore its feasibility in the analysis of large-scale particle regional structures.

3.2. Evaluation methods

To determine which interpolation approach is suitable for the time-coherent analysis of large-scale particle structure by objective image quality evaluation, we adopt two evaluation methods, involving the peak signal-to-noise ratio (PSNR) and structural similarity (SSIM).

The PSNR is defined as:

$$PSNR(f,g) = 10\lg\frac{255^2}{MSE(f,g)} \tag{3}$$

where

$$MSE(f,g) = \frac{1}{MN}\sum_{i=1}^{M}\sum_{i=1}^{N}(f_{ij} - g_{ij})^2, \tag{4}$$

where f and g represent two images with size of $M * N$, which together with i and j, indicate the pixel position (Hore and Ziou, 2010). The critical part of PSNR is the mean square error (MSE). The higher the PSNR value is, the better the image.

The SSIM is calculated by brightness and contrast, which offsets the flaw of PSNR and gets close to the human senses (Wang et al., 2003). The larger the value, the more similarities between two input images. The corresponding formula is

$$SSIM(f,g) = l(f,g)c(f,g)s(f,g), \tag{5}$$

where l, c, s respectively represent luminance, contrast, and structure.

4. Visual analytics approach

We present PRSVis as an auxiliary system to analyze the time-series particle-based region structure. This section is composed of two themed parts. First, we demonstrate a holistic view of the proposed analysis, correlated system, and functional modules. Furthermore, we supply the adopted methods in PRSVis, containing the rendering method, selection algorithm, and interpolation approach.

4.1. System overview

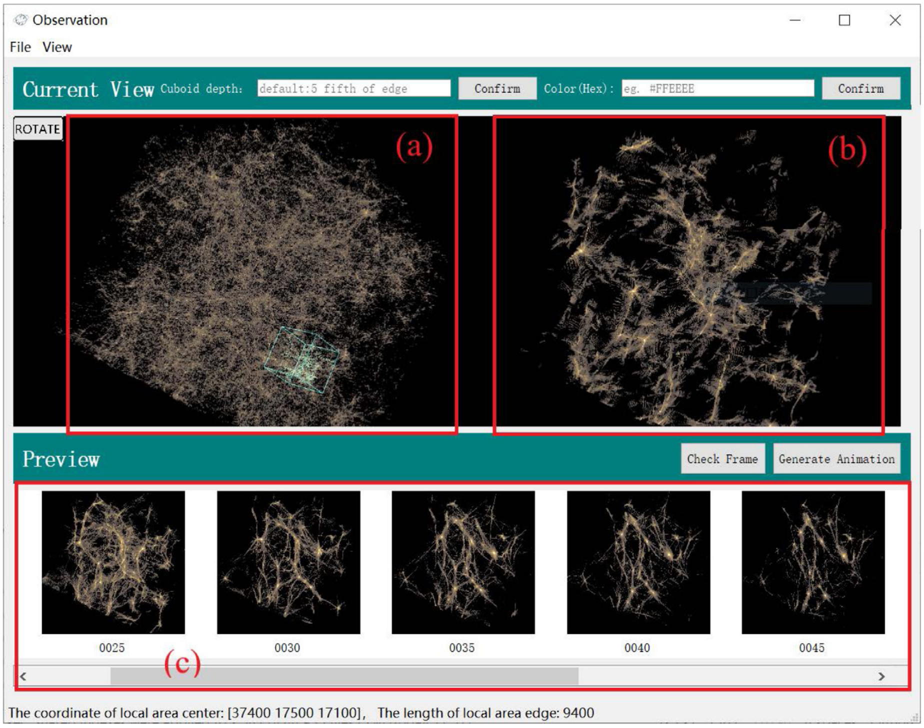
The scheme overview of the interactively time-coherent analysis is shown in Fig. 1. Initially, we obtain the numerical simulation output files. To render all particles, we extract the necessary data from these timestep files. For example, one file contains simulation parameters, particle ID, positions, velocity, and other particle information. We extract the particle ID and position, and then sort them by particle ID. When these data have been processed, we perform the corresponding 3D global rendering, which shows all particle distributions. Following this display, we select one region of the global rendering. In this step, the selection algorithm works to confirm the local cube zone. Then, we demonstrate the previews of the cube pattern with a certain time interval to generally browse the variation trends of the structure. Finally, we generate the smooth animation based on the preview images.

To help astronomers coherently analyze the structure of the universe, we design the interactive visualization system, called PRSVis. The function of PRSVis includes visualizing the global simulation scene, selecting local space, and showing local structure change. The main interface (see Fig. 2) is divided into two parts: selected timestep display and preview of the latter timesteps. The system shows the 3D global visualization of cosmic simulation on the left of the center view. The rendering on the right of the center view is the 3D local structure which is the same as the selected area in the left global rendering. At the bottom, PRSVis displays several previews of the partial region in the following timesteps. Users click on the button "Generate Animation" to score the final time-series animation.

We take the interactive visualization into account. The PRSVis contains several human–computer interaction parts. Users first select the input file, which is processed into the format:

$$\{id_1, position_1, id_2, position_2, \ldots, id_n, position_n\}.$$

After global rendering, users draw a lasso on the screen to select the local region. As the final area is a cube, we apply a selection



The coordinate of local area center: [37400 17500 17100], The length of local area edge: 9400

Fig. 2. The universe structure analysis consists of the following steps: (a) 3D rendering of the input file. (b) Enlarged local rendering of the selected region, which is the blue box on the left side. (c) After confirming the view angle, this region demonstrates previews of the selected cube.

algorithm to confirm the final region. In addition, users have two choices to rotate, amplify, shrink and translate the 3D rendering, first for the region selected, and the other for the animation view confirmed. Furthermore, the corresponding buttons have to be clicked before every step of the analysis is conducted.

4.2. Adopted algorithms

We apply three main algorithms in our system. To show the rendering, we utilize the improved tetrahedral tessellation approach. Once the selected time arrives with the lasso, the selection algorithm runs. In the final step of the analysis, the system generates animation using the deep-learning-assisted method.

4.2.1. Rendering method

We adopt an improved tetrahedral tessellation approach (Kaehler et al., 2012) to render. This method, which regards every particle as one vertex of tetrahedrons that deform over time. Finally, all the tetrahedrons are superimposed to show a clear structure. This approach not only visualizes the large-scale universe structures more clearly but also improves calculation speed.

4.2.2. Selection algorithm

To score the final region, we first confirm the 2D range and expand it to the 3D area. The selection algorithm first records the maximum and minimum values of the points over the lasso track to determine the largest square. We use the X-axis and Y-axis to represent the screen. The calculation formulas for the center and edge of the square are:

$$center(x) = min(X) + \frac{max(X) - min(X)}{2} \tag{6}$$

$$center(y) = min(Y) + \frac{max(Y) - min(Y)}{2} \tag{7}$$

$$edge = max(max(X) - min(X), max(Y) - min(Y)) \tag{8}$$

In these equations, X and Y respectively represent the position array of the X-axis and Y-axis. The algorithm first expands the square into a rectangular solid in the perpendicular-to-the-screen direction (Z-axis). We combine our experience and the particle distribution to confirm the depth of the cube. The processes are as follows: the cuboid is equidistantly divided into several smaller cuboids in the Z-axis. We set the default depth of the small cuboids as 1/5 of the square's edge, which can be adjusted in our system. To find the densest cuboid, we calculate the quality and volume of every small cuboid. Based on the center of this cuboid, the algorithm constructs a new cube with sides that are the same as the length of the beginning square. This cube is the object of rendering again.

4.2.3. Interpolation approach

Generating a smooth time-series animation in the time-coherent analysis is a critical task. We interpolate frames utilizing the Super SloMo method to score the cartoon.

Super SloMo is a DCNN approach that applies an end-to-end CNN model. It combines traditional algorithms to obtain the superior optical flow. The optical flow is the instantaneous velocity of the pixel movement, which can be understood as the instantaneous grayscale change of the pixel. In the structure evolution animation, every particle map in the visualization appears as a bright dot. If many particles gather in one place, then an eye-catching spot appears. Thus, we can deduce the particle movement based on the change in brightness in the optical flow.

The framework of Super SloMo is inspired by U-Net (Ronneberger et al., 2015), which has performed well in previous experiments. Super SloMo is time-independent, which indicates that it is suitable to control the interpolation frame number and time interval. Furthermore, this model considers object motion and blocks by adjusting the loss function. This refinement effectively eliminates artifacts. As Super SloMo is predicted well in our experiments, we apply it to our time-series analytics system.

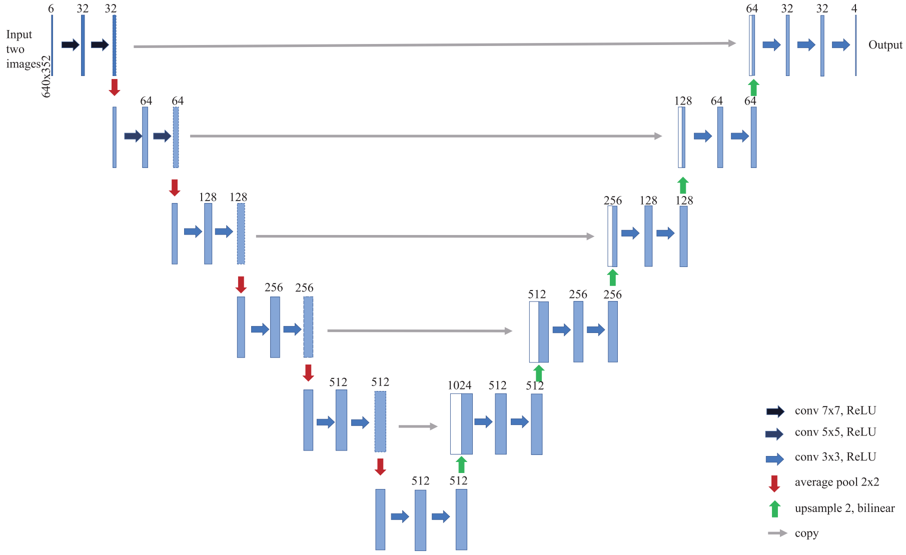


Fig. 3. In the network architecture of optical flow computation, the network first downsamples the input, which are two cropped images, through several sections, involving convolution and pooling. Following this step, the network upsamples the middle result to output the bi-directional optical flow.

In the overall application of Super SloMo, the inputs are discrete sequence frames, which are the critical components of animations. The interpolated frames are the outputs of this model. In addition, we collect numerous sequential timestep snapshots from universe simulation to train our model. These snapshots are the universe scene renderings.

The network of Super SloMo is divided into two parts: flow calculation and flow interpolation. Fig. 3 shows the structure of the first part. This structure is displayed as "U" symmetry, where the front of "U" is the encoder and the latter is the decoder. The specific steps are the following:

Step 1 Inputting of two cropped pictures.

Step 2 Passing through a block, which contains two convolution layers and a Leaky ReLU filter layer. The kernel size of convolution is 7*7 for large motion capture.

Step 3 Downsampling by the average pooling and passing a similar block. This block includes convolution layers whose kernel size is 5*5.

Step 4 Step 3 is repeated four times with the convolution kernels of size 3*3.

Step 5 Upsampling with scaling 2 using bilinear interpolation and going through two 3*3 convolution layers together with activating functions five times.

Step 6: Passing the final convolution layer and activating the output by Leaky ReLU.

Step 7: Outputting the bi-directional optical flows.

We use the forward and backward optical flow to estimate the bi-directional optical flows of the interpolated frames. The corresponding equations are

$$\widehat{F}_{t\to0} = -(1-t)tF_{0\to1} + t^2F_{1\to0} \tag{9}$$

$$\widehat{F}_{t\to1} = (1-t)^2F_{0\to1} - t*(1-t)F_{1\to0}, \tag{10}$$

where t is time and F is optical flow.

The architecture of the flow interpolation is the same as the flow calculation except for the inputs and outputs. This architecture contains eight inputs: input frames I_0 and I_1, bi-directional optical flows between inputs $F_{0\to1}$ and $F_{1\to0}$, and bi-directional optical flows of interpolated frames $\widehat{F}_{t\to0}$ and $\widehat{F}_{t\to1}$, together with the two images calculated by the warped function $g(I_0, F_{t\to0})$ and $g(I_1, F_{t\to1})$. This network outputs the refined flows and soft visibility maps, which show the occlusion. We obtain the final interpolated frames through these processes. The loss function of Super SloMo is defined as

$$l = \lambda_r l_r + \lambda_p l_p + \lambda_w l_w + \lambda_s l_s, \tag{11}$$

where the λ is the weight and l denotes loss. The four parts in the formulation represent loss evaluating the rebuilding quality of interpolated frames, warping loss modeling the quality of the estimated optical flows, perceptual loss preserving the details together with the sharpness of edge, and smoothness loss appraising the smoothness between close pixels.

5. Experiments and results

In this section, we show the application results of the Super SloMo in the analysis of universal structure formation. We first introduce the datasets used in the network training and how to train the network. Then we describe the test environment. In our experiment, to determine the most suitable interpolation approach, we compare four interpolation methods, namely, particle-position-based, particle-acceleration-based, phase-based, and Super SloMo. We illustrate the basic theory of these methods and the reason for choosing them in 3.1. Furthermore, we show the comparison results by quantitative evaluation and intuitive visual perception.

5.1. Datasets and implementation details

Experiment datasets are generated by the GADGET-2 (Springel, 2005), which is a piece of code designed for cosmological N-body simulations. This freely available code can be run massively parallel computers with distributed memory, utilizes an explicit communication model and a standard MPI interface, which fit our experiment requirements. The simulation contains a set of control parameters. For example, the source code contains several different types of parameter files, like cluster and galaxy. Users confirm the initial parameters according to the simulation content. In the galaxy parameter file, some parameters control the name of

Table 1
Result on PSNR & SSIM.

Method	PSNR			
Periods	0500–0600		1000–1100	
Frame number	4 frames	9 frames	4 frames	9 frames
Particle-acceleration-based	25.0476	25.2357	27.7599	26.7413
Particle-position-based	26.3043	26.9604	26.4698	27.1377
Phase-based	25.7102	25.7874	26.1325	26.9054
Super SloMo	25.8387	26.0138	26.6007	27.3375
Method	SSIM			
Periods	0500–0600		1000–1100	
Frame number	4 frames	9 frames	4 frames	9 frames
Particle-acceleration-based	88.04%	88.29%	91.63%	90.27%
Particle-position-based	89.92%	90.81%	89.59%	90.80%
Phase-based	84.16%	84.88%	86.20%	87.43%
Super SloMo	87.78%	88.06%	89.47%	90.15%

Table 2
The result of time evaluation.

Periods	0500–0600		1000–1100	
Frame number	4 frames	9 frames	4 frames	9 frames
Particle-acceleration-based	507 s	1108 s	509 s	1129 s
Particle-position-based	498 s	1055 s	529 s	1154 s
Phase-based	44.952 s	76.993 s	43.677 s	76.423 s
Super SloMo	7.244 s	8.426 s	8.335 s	9.131 s

relevant output files. other code options parameters decide the snapshot format, whether to integrated the comoving frame, and whether to choose period boundaries. The timestep is a time sign that match to snapshot files. The output folder contains information files that describe simulation situation, parameter files, and all the snapshot files that save the basic parameters and the details of every particle, like particle id, position, and velocity. To generate enough experimental data, we adjusted partial parameters of GADGET-2 and built different simulations of universe evolution. In the m ain experiment, we set the simulation space as a cube, whose sides are 50 Mpc. We use 2^{21} particles to simulate the evolution of cosmic structure.

Training datasets for Super SloMo are collected from simulation visualization conducted with GADGET. The Super SloMo requires sequence inputs such as videos frames. Therefore, we set different simulation parameters to collect movies from several simulations. The parameters contain the particle number, simulation scale, and color mapping table. Then, we convert these videos into a series of images and cut them into a uniform size. Pictures from the same video are divided into training and test sets by a ratio of 9:1. When training Super SloMo, we used Adam to optimize together with a self-adapting learning rate from $1e^{-4}$ to $1e^{-6}$ according to the iterations.

In our **experiment environment**, we used GeForce GTX 1080 Graphics Card (NVIDIA Corporation) to train the deep learning network. We compare the performance of different methods using processor $Intel(R)Xeon(R)W - 2223CPU@3.60GHz3.60GHz$.

5.2. Quantitative evaluation

We compared the two linear particle-based interpolation algorithms, phase-based interpolation algorithm modified by Meyer, and the Super SloMo by our datasets mentioned in Section 5.1. The experiment quantitatively evaluated the image quality and interpolation time of the interpolation outcome.

We divided the output of cosmic simulation into 1501 timestep files, from the birth of the universe to the present. Due to the distinctly different stage status of the cosmic evolution, we extracted fragments from several sections. In addition, as the early stage of formation is unstable, we test it on the later periods. To guarantee the reliability of the experimental conclusion, we tested multifacetedly by choosing two periods of time and separately interpolating four frames and nine frames from timestep 0500 to 0600 and from timestep 1000 to 1100. Furthermore, we analyze the universe structure formation after selecting a region. Therefore, the comparison objects are the selected region from those generated frames and the source images generated by original data.

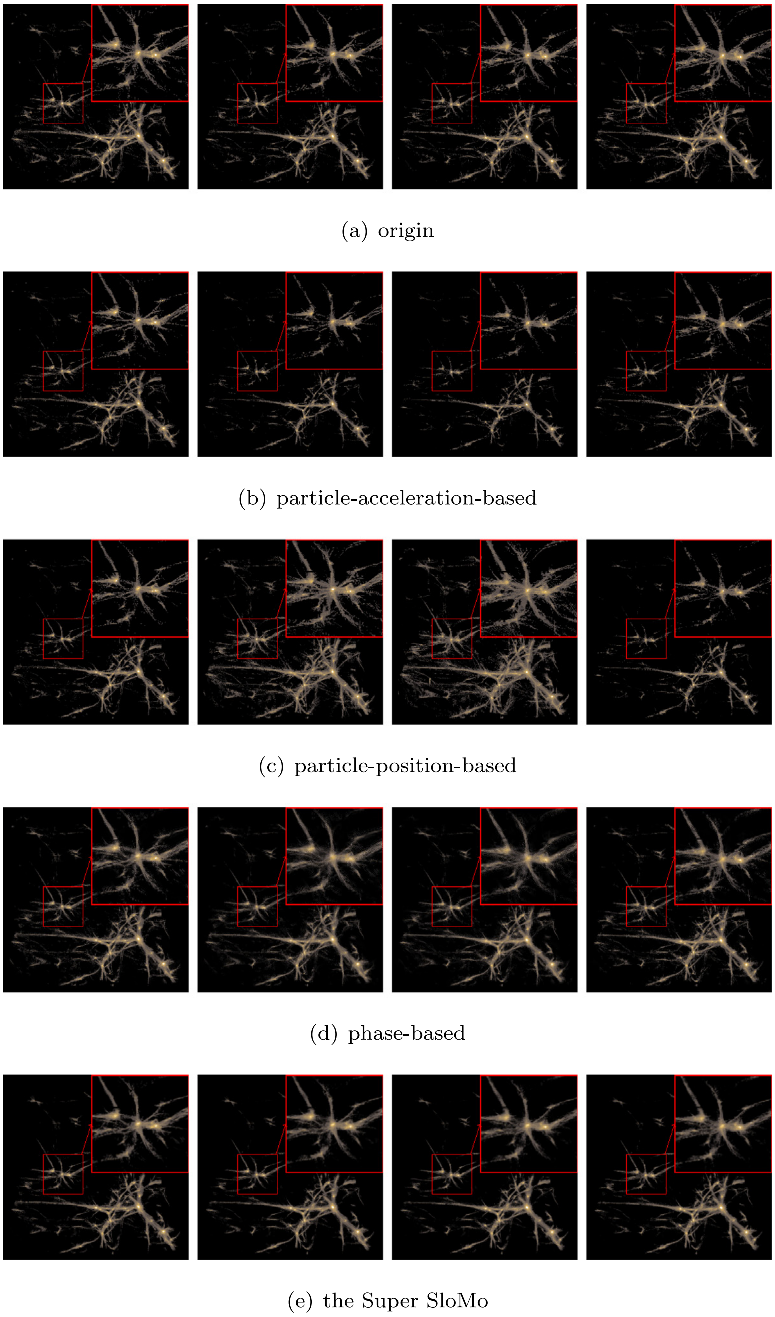
Table 1 shows the average results of the generated frames in two stages evaluated by PSNR and SSIM. When we focus on one row, the values are similar in the same periods. Furthermore, all the interpolation algorithms performed better over phase 1000–1100 than 0500–0600. The results are reasonable, and are related to the process of cosmic evolution. The dark matter particles distribute uniformly at the beginning of universe simulation and then move in a dispersed manner. Over time, these particles move more slowly and tend to be stable. The bigger the number of timesteps is, the less change in the position of the particles. The interpolation algorithms perform better when two more similar images are input.

Besides, Table 1 shows the evaluation results of different algorithms from the column perspective. The structure of the universe changes more sharply in the phase 0500–0600 than the phase 1000–1100. The two particle-based methods perform similarly, although the particle-position-based method is superior. The particle-acceleration-based method performs much better when interpolated between 1000–1100 than between 0500–0600. We analyzed the results of every interpolated frame, and we found that the variances of PSNR results of particle-based methods are much larger than the those of other two methods. When 9 frames are interpolated between 0500–0600, a range underwent a greater change than 1000–1100, and the PSNR results of particle-based methods fluctuate much more than the others do. Super SloMo keeps the better standard in 0500–0600 and 1000–1100, which indicates that it performed more stably in this experiment. Its values are basically close to the best value at every column when evaluating based on PSNR and SSIM. Furthermore, the Super SloMo method is superior to the phase-based method, close to the particle-position-based method, even sometimes outperforms it. The experiment results and actual phenomenon indicate that the Super SloMo is more flexible than the other two methods. In addition, the Super SloMo performed more stably by standard deviation evaluation, which reflects the degree of dispersion between individuals in a group.

5.3. Visual performance

Table 2 describes the time cost of interpolation by the four algorithms. The time that the particle-based methods spend is proportional to the interpolated frame number, limiting the interpolation number to a certain degree. In addition, the particle-based methods interpolate frames in three dimensions, which need the extra process of transforming the 3D volume to the 2D figure. For each frame, this process costs around three minutes. These problems hinder the quick interaction. The phase-based algorithm is around ten times faster than the particle-based method, although it is also related to the interpolation number. The Super SloMo, which benefits from pre-training, needs only several seconds to generate new images. The speeds are vary distinctly. In addition, the particle-based method reads all particles, which puts enormous pressure on input/output (I/O) operation. The other two approaches handle images instead, thereby reducing the strain of the random access memory (RAM) and I/O.

In Fig. 4, we demonstrate the local images generated by the four ways and the original images in the periods 0500–0600. Every row shows a variation trend from the timestep 0500 to 0600.



(a) origin

(b) particle-acceleration-based

(c) particle-position-based

(d) phase-based

(e) the Super SloMo

Fig. 4. Samples of interpolation results. Fig. 4 shows the source images and the general map of the images generated using four interpolation methods. Every row represents the results of one method while every column shows the difference between the source images. The outputs of interpolation at the same timestep, from left to right, are 0520, 0540, 0560 and 0580. In addition, the red box in the upper-right corner of every image is the larger version of the red box on the lower left-center.

In the first row, the source images show two bright dots spread apart in the left of the larger red box. The images generated by the particle-acceleration-based method ignore the edge lines and only preserve the cores. By contrast, the particle-position-based method generates a few artifacts. Particles in the middle images disperse, and thus the whole picture is blurred. The track of the particles is difficult to detect, let alone linearly define. However, the more complex the formula is, the larger the calculation is. The image produced by the phase-based approach is unambiguous at first glance. However, from the column of timestep 0560, the images created by the last two methods are different. The source image contains two small bright yellow spots in the partial enlargement, while the phase-based image makes the center spot disappear. In general, Super SloMo generates the most similar images from the naked eye, which retains the cores and generates a similar filamentous structure.

6. Discussion

In this section, we summarize the effectiveness of applying Super SloMo to our system, the requirement and the feedback of the domain experts, as well as the limitations of the system integrated with the Super SloMo.

Super SloMo is designed for video interpolations, which results in high-quality estimation. It considers the object motion, which could correspond to particle movement. Furthermore, Super SloMo is based on an end-to-end deep learning network,

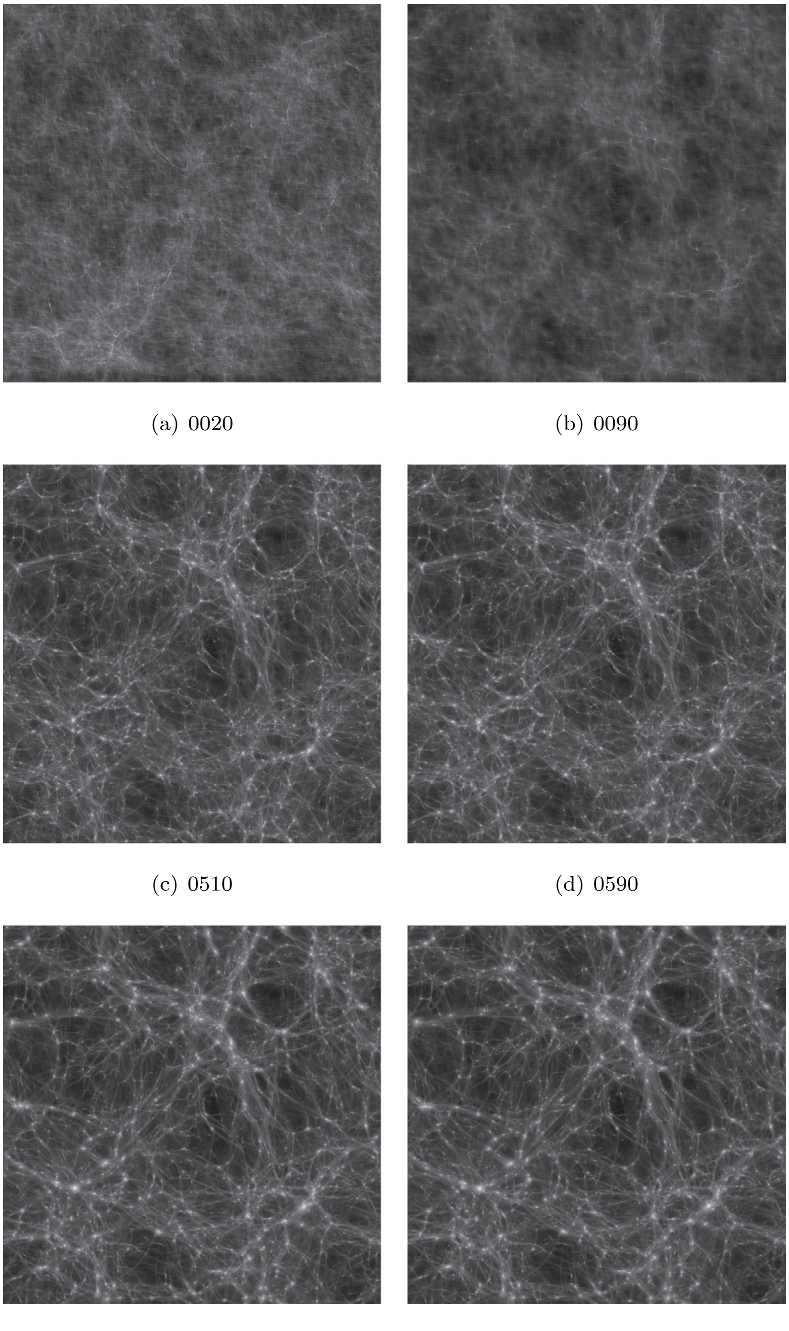

(a) 0020
(b) 0090
(c) 0510
(d) 0590
(e) 1010
(f) 1090

Fig. 5. The six pictures are from the evolution of the universe simulation, according to the data generated by GADGET-2. This simulation contains 1501 snapshots, and these pictures are on behalf of the early, middle and late stages.

which has a process of training sets, to ensure better performance. We choose the training sets gathered from astronomy, which makes the network respond in a way that is close to our needs. We evaluate its effectiveness by comparing experiments from both quantity and visual quality. In addition, the analytic system was integrated with Super SloMo, which smoothens the pipeline from finding one local area to observing its change in time.

We consider the requirement of the astronomers for the simulation analysis. The first task requires an overview of the simulation data, which provides insight into the global distribution (T1). In addition, to observe the evolution of the subhalo, scientists have to flexibly select the focus region (T2) and clearly show the high-resolution local region (T3), and show the temporal change (T4). For these requirements, we designed the PRSVis system, which first allows users to transform the data file into volume visualization, showing the space situation of the simulation data. Then, the system allows users to select a region from the global volume and automatically show the local volume. In this step, the user outlines the region freely, and the system is calculated by density to obtain the maximum density cube. The point of view deeply influences the understanding of the structural evolution. If we observe a dot gradually brighten, then we may see a merge occurring in the third dimension. Thus we provide the rotatable local cube that allows the user to adjust the angle and obtain evolution animation at a suitable angle.

The feedback from the domain expert indicates that visualizing the feature structure helps experts to understand the ptime cost of interporocess of its formation. Our system allows visualization of a small-range region that is interactively selected by users, which is meaningful for data exploration from large-scale simulation of the universe and for finding the distribution and evolution of the dark halos. The interactive selection and regional evolution demonstration help scientists analyze a set of key events in the target area, which happens in the evolution of black halos. These modules benefit the confirmation and region selection of key parameters in some analysis tasks, such as re-simulation and merge tree.

Although our evaluations and expert feedback demonstrated the effectiveness of applying Super SloMo in the analysis of the large-scale structure of the universe, our work has limitations. First, although Super SloMo calculates the interpolated frames quickly, which is the foundation of high interaction, sometimes its outcome cannot obtain the best quality among these comparison methods. A possible improvement is to make the network learn the particle number and then use this parameter to adjust the local brightness of interpolated images. Second, although we have designed a complete system to freely explore the evolution of local regions, we have not given the choice to goal-oriented selection. Thus, more optional interactive operations should be considered.

7. Conclusion

To prove the feasibility of a deep-learning-assist interpolation method in the structural analysis of large-scale particle regions, we apply this method in analyzing the cosmic structure formation and then evaluate the results from objection and subjection aspects. As shown in Tables 1 and 2, Super SloMo has a high computation speed and a stable performance, which is close to the best in all quality tests. As shown in Fig. 4, Super SloMo generates interpolation images that are consistent with human perception. These experiments suggest that Super SloMo is suitable in the particle region structure. In addition, to assist scientists in coherently analyzing and observing the details of the structure, we propose an interactive visualization tool, PRSVis. As shown in Fig. 2, users begin with an N-body numeric simulation rendering, interactively select a local area, and observe the subsequent changes to the structure from the bottom previews.

CRediT authorship contribution statement

Yihan Zhang: Conceptualization, Data curation, Formal analysis, Investigation, Visualization, Writing – original draft. **Guan Li:** Resources, Methodology, Software, Validation. **Guihua Shan:** Supervision, Project administration, Writing – review & editing, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Ethical Approval

This study does not contain any studies with human or animal subjects performed by any of the authors. All data used in the study are taken from public databases that were published in the past.

Acknowledgment

This work is supported by the National Key Research and Development Program of China (2020YFB0204802).

Appendix. Examples of cosmological simulation data

See Fig. 5.

References

Ahrens, J., Jourdain, S., O'Leary, P., Patchett, J., Rogers, D.H., Petersen, M., 2014. An image-based approach to extreme scale in situ visualization and analysis. In: SC'14: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. IEEE, pp. 424–434.

Baker, S., Matthews, I., 2004. Lucas-kanade 20 years on: A unifying framework. Int. J. Comput. Vis. 56 (3), 221–255.

Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M.J., Szeliski, R., 2011. A database and evaluation methodology for optical flow. Int. J. Comput. Vis. 92 (1), 1–31.

Barnes, J., Hut, P., 1986. A hierarchical O (N log N) force-calculation algorithm. Nature 324 (6096), 446–449.

Bertschinger, E., 1998. Simulations of structure formation in the universe. Annu. Rev. Astron. Astrophys. 36 (1), 599–654.

Black, M.J., Anandan, P., 1991. Robust dynamic motion estimation over time. In: Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 296–302.

Black, M.J., Anandan, P., 1996. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. Comput. Vis. Image Underst. 63 (1), 75–104.

Black, M.J., Jepson, A.D., 1996. Estimating optical flow in segmented images using variable-order parametric models with local deformations. IEEE Trans. Pattern Anal. Mach. Intell. 18 (10), 972–986.

Boykov, Y., Veksler, O., Zabih, R., 2001. Fast approximate energy minimization via graph cuts. IEEE Trans. Pattern Anal. Mach. Intell. 23 (11), 1222–1239.

Darabi, S., Shechtman, E., Barnes, C., Goldman, D.B., Sen, P., 2012. Image melding: Combining inconsistent images using patch-based synthesis. ACM Trans. Graph. 31 (4), http://dx.doi.org/10.1145/2185520.2185578.

Davis, M., Efstathiou, G., Frenk, C.S., White, S.D., 1985. The evolution of large-scale structure in a universe dominated by cold dark matter. Astrophys. J. 292, 371–394.

Davis, A., Rubinstein, M., Wadhwa, N., Mysore, G.J., Durand, F., Freeman, W.T., 2014. The visual microphone: Passive recovery of sound from video. ACM Trans. Graph. 33 (4), http://dx.doi.org/10.1145/2601097.2601119.

Einstein, A., 1919. What Is the Theory of Relativity?. Springer.

Gamow, G., 1948. The evolution of the universe. Nature 162 (4122), 680–682.

Glocker, B., Paragios, N., Komodakis, N., Tziritas, G., Navab, N., 2008. Optical flow estimation with uncertainties through dynamic MRFs. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8.

Halpern, P., Tomasello, N., 2016. Size of the observable universe. Adv. Astrophys. 1 (3), 135–137.

Hore, A., Ziou, D., 2010. Image quality metrics: PSNR vs. SSIM. In: 2010 20th International Conference on Pattern Recognition. pp. 2366–2369.

Janai, J., Guney, F., Wulff, J., Black, M.J., Geiger, A., 2017. Slow flow: Exploiting high-speed cameras for accurate and diverse optical flow reference data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3597–3607.

Jiang, L., Kashikawa, N., Wang, S., Walth, G., Ho, L.C., Cai, Z., Egami, E., Fan, X., Ito, K., Liang, Y., et al., 2021. Evidence for GN-z11 as a luminous galaxy at redshift 10.957. Nat. Astron. 5 (3), 256–261.

Jiang, H., Sun, D., Jampani, V., Yang, M., Learned-Miller, E., Kautz, J., 2018. Super SloMo: High quality estimation of multiple intermediate frames for video interpolation. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, pp. 9000–9008.

Kaehler, R., Hahn, O., Abel, T., 2012. A novel approach to visualizing dark matter simulations. IEEE Trans. Vis. Comput. Graphics 18 (12), 2078–2087. http://dx.doi.org/10.1109/TVCG.2012.187.

Kageyama, A., Sakamoto, N., 2020. 4D street view: a video-based visualization method. PeerJ Comput. Sci. 6, e305.

Kuhlen, M., Vogelsberger, M., Angulo, R., 2012. Numerical simulations of the dark universe: State of the art and the next decade. Phys. Dark Univ. 1 (1–2), 50–93.

Lei, C., Yang, Y., 2009. Optical flow estimation on coarse-to-fine region-trees using discrete optimization. In: 2009 IEEE 12th International Conference on Computer Vision. pp. 1562–1569.

Lemaître, G., 1927. Un Univers homogène de masse constante et de rayon croissant rendant compte de la vitesse radiale des nébuleuses extra-galactiques. In: Annales de la SociÉTÉ Scientifique de Bruxelles, Vol. 47. pp. 49–59.

Lempitsky, V., Roth, S., Rother, C., 2008. FusionFlow: Discrete-continuous optimization for optical flow estimation. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8.

Liu, Z., Yeh, R.A., Tang, X., Liu, Y., Agarwala, A., 2017. Video frame synthesis using deep voxel flow. In: 2017 IEEE International Conference on Computer Vision. ICCV, pp. 4473–4481. http://dx.doi.org/10.1109/ICCV.2017.478.

Lu, Q., Xu, N., Fang, X., 2016. Motion-compensated frame interpolation with multiframe-based occlusion handling. J. Disp. Technol. 12 (1), 45–54.

Lucas, B.D., 1981. An iterative image registration technique with an application to stereo vision (DARPA). Proc Ijcai 81 (3), 674–679.

MARKMAN, E., 1979. Thinking in perspective: Critical essays in the study of thought processes. Psyccritiques 24 (9).

Meyer, S., Wang, O., Zimmer, H., Grosse, M., Sorkine-Hornung, A., 2015. Phase-based frame interpolation for video. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1410–1418.

Niklaus, S., Mai, L., Liu, F., 2017. Video frame interpolation via adaptive convolution. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition. CVPR, pp. 2270–2279.

Novikov, I.D., 1983. Evolution of the Universe. University Press, Cambridge.

Ratra, B., Vogeley, M.S., 2008. The beginning and evolution of the universe. Publ. Astron. Soc. Pac. 120 (865), 235.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 234–241.

Sadek, R., Ballester, C., Garrido, L., Meinhardt, E., Caselles, V., 2012. Frame interpolation with occlusion detection using a time coherent segmentation. In: VISAPP (2). pp. 367–372.

Seitz, S.M., Baker, S., 2009. Filter flow. In: 2009 IEEE 12th International Conference on Computer Vision. IEEE, pp. 143–150.

Somerville, R.S., Davé, R., 2015. Physical models of galaxy formation in a cosmological framework. Annu. Rev. Astron. Astrophys. 53, 51–113.

Soomro, K., Zamir, A.R., Shah, M., 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402.

Springel, V., 2005. The cosmological simulation code gadget-2. Mon. Not. R. Astron. Soc. 364 (4), 1105–1134. http://dx.doi.org/10.1111/j.1365-2966.2005.09655.x, arXiv:https://academic.oup.com/mnras/article-pdf/364/4/1105/18657201/364-4-1105.pdf.

Steinhardt, P.J., Turok, N., 2002. Cosmic evolution in a cyclic universe. Phys. Rev. D 65 (12), 126003.

Su, S., Delbracio, M., Wang, J., Sapiro, G., Heidrich, W., Wang, O., 2017. Deep video deblurring for hand-held cameras. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition. CVPR, pp. 237–246. http://dx.doi.org/10.1109/CVPR.2017.33.

Sun, J., Zheng, N.-N., Shum, H.-Y., 2003. Stereo matching using belief propagation. IEEE Trans. Pattern Anal. Mach. Intell. 25 (7), 787–800.

Trobin, W., Pock, T., Cremers, D., Bischof, H., 2008. An unbiased second-order prior for high-accuracy motion estimation. In: Proceedings of the 30th DAGM Symposium on Pattern Recognition. pp. 396–405.

Wadhwa, N., Rubinstein, M., Durand, F., Freeman, W.T., 2013. Phase-based video motion processing. ACM Trans. Graph. 32 (4), http://dx.doi.org/10.1145/2461912.2461966.

Wang, Z., Simoncelli, E., Bovik, A., 2003. Multiscale structural similarity for image quality assessment. In: The Thrity-Seventh Asilomar Conference on Signals, Systems Computers, 2003, Vol. 2. pp. 1398–1402 Vol.2. http://dx.doi.org/10.1109/ACSSC.2003.1292216.

Warren, D., Strelow, E., 1985. Electronic Spatial Sensing for the Blind: Contributions from Perception, Rehabilitation, and Computer Vision. In: Nato Science Series E:, Springer Netherlands, URL https://books.google.com.hk/books?id=-I_Hazgqx8QC.

Wedel, A., Pock, T., Braun, J., Franke, U., Cremers, D., 2008. Duality TV-L1 flow with fundamental matrix prior. In: 2008 23rd International Conference Image and Vision Computing New Zealand. pp. 1–6.

Werlberger, M., Pock, T., Unger, M., Bischof, H., 2011. Optical flow guided TV-l 1 video interpolation and restoration. In: International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition. Springer, pp. 273–286.

Yamaoka, Y., Hayashi, K., Sakamoto, N., Nonaka, J., 2019. In situ adaptive timestep control and visualization based on the spatio-temporal variations of the simulation results. In: Proceedings of the Workshop on in Situ Infrastructures for Enabling Extreme-Scale Analysis and Visualization. pp. 12–16.