



OPEN ACCESS

Splotch: visualizing cosmological simulations

To cite this article: K Dolag *et al* 2008 *New J. Phys.* **10** 125006

View the [article online](#) for updates and enhancements.

You may also like

- [COSMOLOGICAL SIMULATIONS OF THE INTERGALACTIC MEDIUM EVOLUTION. II. GALAXY MODEL AND FEEDBACK](#)

Benoit Côté, Hugo Martel and Laurent Drissen

- [Bayesian inference using JET's microwave diagnostic system](#)

S. Schmuck, J. Svensson, L. Figini et al.

- [Modelling the temperature evolution of bone under high intensity focused ultrasound](#)

H M M ten Eikelder, D Bošnjaki, A Elevelt et al.

Splotch: visualizing cosmological simulations

K Dolag^{1,3}, M Reinecke¹, C Gheller² and S Imboden²

¹ Max-Planck-Institut für Astrophysik, PO Box 1317, D-85741 Garching, Germany

² High Performance System Division, CINECA, I-40127, Bologna, Italy

E-mail: kdolag@mpa-garching.mpg.de

New Journal of Physics **10** (2008) 125006 (18pp)

Received 6 May 2008

Published 1 December 2008

Online at <http://www.njp.org/>

doi:10.1088/1367-2630/10/12/125006

Abstract. We present a light and fast, publicly available, ray-tracer Splotch software tool which supports the effective visualization of cosmological simulations data. We describe the algorithm it relies on, which is designed in order to deal with point-like data, optimizing the ray-tracing calculation by ordering the particles as a function of their ‘depth’, defined as a function of one of the coordinates or other associated parameters. Realistic three-dimensional impressions are reached through a composition of the final colour in each pixel properly calculating emission and absorption of individual volume elements. We describe several scientific as well as public applications realized with Splotch. We emphasize how different datasets and configurations lead to remarkably different results in terms of the images and animations. A few of these results are available online.

³ Author to whom any correspondence should be addressed.

Contents

1. Introduction	2
2. Splotch	3
2.1. Physical motivation of the algorithm	3
2.2. Implementation of the algorithm	5
2.3. Public version of the algorithm	5
2.4. Performance of the algorithm	5
3. Application to cosmological simulations	6
3.1. Visualizations using flat projection	6
3.2. Animations of evolving galaxy clusters	8
4. The 4D universe	11
4.1. Splotch usage and stereographic rendering	11
4.2. The Big Bang	13
4.3. The evolution of the universe	13
4.4. The galaxy	15
5. Conclusions	17
Acknowledgments	18
References	18

1. Introduction

According to the standard theory of the *Big Bang*, our universe came into existence as a singularity around 13.7 billion years ago. After its initial appearance, it inflated from a very dense, hot and homogeneous state to the expanded, cooled and a very structured state of our current universe. During its evolution, planets, stars, galaxies and all the different objects and structures that we observe today have formed as a result of a fine interplay of gravitational forces with a number of different physical processes, like fluid dynamics effects (shock waves and turbulence), magnetic fields, star formation and associated feedback (e.g. supernova explosions). Cosmologists adopt numerical simulations as an effective instrument for describing, investigating and understanding the evolution of such a complicated system within the framework of the expanding space–time of the universe. There are various numerical methods—grid- or particle-based—to perform such simulations. For a recent review on such numerical simulation methods within the cosmological context, see Dolag *et al* (2008).

The resulting data is huge and complex and requires suitable and effective tools to be inspected and explored. Visualization represents the most immediate and intuitive way of analysing and exploring data and of understanding the results. Larger datasets require an enormous computational effort to be systematically studied and more and more complex and expensive algorithms to identify patterns and features. Visualization allows one to focus easily on subsamples and features of interest and to detect correlations and special characteristics of data. This is not a substitute for systematic analysis, but it is a fundamental support for accelerating and simplifying the cognitive process.

Visualization is also an effective instrument for introducing the general public, especially young people, to even the most complicated and innovative aspects and concepts of science.

The opportunity to use three-dimensional (3D) digital visualization and virtual reality facilities further enhances the impact of the communication and divulgence and pushes towards a new approach to cultural and scientific heritage.

The availability of tools for visualizing scientific data in a comprehensible, self-describing, rich and appealing way is therefore crucial for researchers and for the general public, both for creating knowledge and for disseminating it.

Such tools must be able to deal with a huge data volume, possibly leading to meaningful results in a reasonable time. Furthermore, they must meet the requirements of the particular discipline which produced the data and fit the specific characteristics and peculiarities of the same data. In particular, most of the state-of-the-art cosmological simulations describe the different matter components in the universe (stars, dark matter, gas, etc) as fluid elements (either point-like or on a regular/irregular grid), which must be properly rendered. For the data based on regular grids many visualization packages are available. However, simulations based on a particle-like description of the fluid elements are much more difficult to visualize. There are standard packages like TIPS⁴ or VisIVO⁵ for displaying such data, however, they are not designed to lead to ray-tracing like images, which might give more realistic impressions of 3D structures or even artistic-like impressions for public outreach.

In this paper, we present Splotch, a public ray-tracing software. Splotch is specifically designed to render in a fast and effective way the different families of point-like data, results of a cosmological, but, more generally, astrophysical simulation. Splotch is based on a lightweight and fast algorithm, which will be described in section 2. Section 3 will present some examples where Splotch was used for producing images for scientific publications and animations presented at various conferences. In section 4, we describe a *4D Universe* project where Splotch was used for producing a public outreach movie which will be shown to the public in the virtual reality facility of the brand new Turin Planetarium. We will describe methods used for generating the data used for the Splotch rendering and the technique adopted to create the stereographic version of the movie.

2. Splotch

The underlying visualization algorithm is a derivative of what in general is called *volumetric ray casting*⁶ and uses an approximation of the radiative transfer equation, which—together with the application of perspective—gives the produced images a very realistic appearance, especially when absorption features are present in the cosmological structures which are to be visualized.

2.1. Physical motivation of the algorithm

The rendering algorithm of Splotch is generally designed to handle point-like particle distributions. We especially have *smoothed particle hydrodynamic* (SPH, [Gingold and Monaghan 1977](#), [Lucy 1977](#)) simulations in mind, where the density of the fluid is described by

⁴ <http://www-hpcc.astro.washington.edu/tools/tipsy/tipsy.html>.

⁵ <http://visivo.cineca.it/>.

⁶ http://en.wikipedia.org/wiki/Volume_ray_casting.

spreading tracer particles by a kernel, most commonly the B_2 -spline (Monaghan and Lattanzio 1985)

$$W(x, h) = \frac{8}{\pi h^3} \begin{cases} 1 - 6 \left(\frac{x}{h}\right)^2 + 6 \left(\frac{x}{h}\right)^3 & 0 \leq \frac{x}{h} < 0.5, \\ 2 \left(1 - \frac{x}{h}\right)^3 & 0.5 \leq \frac{x}{h} < 1, \\ 0 & 1 \leq \frac{x}{h}, \end{cases} \quad (1)$$

where h is the local smoothing length, which is typically defined in a way such that every particle overlaps with ≈ 64 neighbours. Therefore, the rendering is based on the following assumptions:

- The contribution to the matter density by every particle can be described as a Gaussian distribution of the form $\rho_p(\vec{r}) = \rho_{0,p} \exp(-r^2/\sigma_p^2)$.⁷ In practice, it is much more handy to have a compact support of the distribution, and therefore the distribution is set to zero at a given distance of $f \cdot \sigma_p$. Following the SPH approach of the original cosmological simulation, we choose f in such a way that $f \cdot \sigma_p$ is related to the smoothing length h , i.e. to fulfil $h \approx f \cdot \sigma_p$. Therefore, rays passing the particle at a distance larger than $f \cdot \sigma_p$ will be practically unaffected by the particle's density distribution.
- We use three 'frequencies' for describing the red, green and blue components of the radiation, respectively. These are treated independently.
- The radiation intensity \mathbf{I} ⁸ along a ray through the simulation volume is modelled by the well-known radiative transfer equation

$$\frac{d\mathbf{I}(\mathbf{x})}{dx} = (\mathbf{E}_p - \mathbf{A}_p \mathbf{I}(\mathbf{x})) \rho_p(\mathbf{x}), \quad (2)$$

which can be found in standard textbooks like Shu (1991). Here, \mathbf{E}_p and \mathbf{A}_p describe the strength of radiation emission and absorption for a given SPH particle for the three rgb-colour components. In general, it is recommended to set $\mathbf{E}_p = \mathbf{A}_p$ which typically produces visually appealing images; for special effects, however, independent emission and absorption coefficients can be used. These coefficients can vary between particles, and are typically chosen as a function of a characteristic particle property (e.g. its temperature, density, etc). The mapping between the scalar property and the three components of \mathbf{E} and \mathbf{A} (for red, green and blue) is typically achieved via a colour look-up table or palette, which can be provided to the ray-tracer as an external file to allow maximum flexibility.

Assuming that absorption and emission are homogeneously mixed, it can be shown that whenever a ray traverses a particle, its intensity change is given by

$$\mathbf{I}_{\text{after}} = (\mathbf{I}_{\text{before}} - \mathbf{E}_p/\mathbf{A}_p) \exp\left(-\mathbf{A}_p \int_{-\infty}^{\infty} \rho_p(\mathbf{x}) d\mathbf{x}\right) + \mathbf{E}_p/\mathbf{A}_p. \quad (3)$$

The integral in this equation is given by $\rho_{0,p} \sigma_p \exp(-d_0^2/\sigma_p^2) \sqrt{\pi}$, where d_0 is the minimum distance between the ray and the particle centre.

Under the assumption that the SPH particles do not overlap, the intensity reaching the observer could simply be calculated by applying this formula to all particles intersecting with the ray, in the order of decreasing distance to the observer. In reality, of course, the particles do overlap, but since the relative intensity changes due to a single particle, it can be assumed to be very small; this approach can nevertheless be used in good approximation.

⁷ Note that the b_2 -spline kernel used in SPH has a shape very similar to the Gaussian distribution.

⁸ Here, we treat all intensities as vectors with r, g and b components.

2.2. Implementation of the algorithm

The rendering algorithm now proceeds as follows:

- A camera position \vec{c} , a look-at point \vec{l} (e.g. the position of the object of interest) and a sky vector \vec{s} are specified by the user. The latter just defines the *up*-direction for the scene, and the first two define the viewing direction \vec{v} , i.e. $\vec{v} = \vec{l} - \vec{c}$. An example of the simulation geometry is given in figure 5 in section 4.1. Now a linear transformation matrix is determined, which maps \vec{c} to the origin and aligns \vec{v} with the positive *z*-axis. This transformation is applied to all particles. Therefore, the ray-tracing can now proceed along the *z*-axis of the new coordinates, which further simplifies the procedure as well as any further operation like perspective or in field-of-view tests.
- A perspective projection is applied, using a user-defined field-of-view angle φ .
- The particles are sorted in the order of descending *z*-coordinate of their centre. Therefore, the ray-tracing (as described in our approximation) can now be achieved by walking the particles and calculating their contribution to the individual pixels of the final image.
- A frame buffer is allocated, which stores an rgb colour triplet for every pixel of the resulting image. This buffer is initialized as black.
- For every particle and every related pixel, equation (3) is used for updating the colour value stored in this pixel.
- Once all particles have been processed, the frame buffer contains the final image, which is written to disk in a standard format (e.g. TGA or JPEG).

2.3. Public version of the algorithm

Besides various internal optimizations, Splotch uses OpenMP⁹ for generating different regions of the image distributed over several cores on shared memory platforms. The algorithm requires 30 bytes per particle, which allows the rendering of more than 70 million particles on a standard PC with 2 GByte of main memory. Therefore, it is applicable to extremely large particle sets, as provided by modern cosmological simulations. The latest version is publically available under GNU General Public License¹⁰ and can be downloaded from <http://www.mpa-garching.mpg.de/~kdolag/Splotch/>. The reading routine is optimized for directly reading the standard output format of P-GADGET2 (Springel 2005) and we strongly encourage users to adapt it to their preferred formats, which can be easily implemented.

2.4. Performance of the algorithm

The CPU time consumption of Splotch to render individual frames depends on the frame size, the number of particles being visualized as well as their apparent size, e.g. over how many pixels do they get spread. For a typical setup, we find $T_{\text{CPU}} \propto n_{\text{pixel}}$ and $T_{\text{CPU}} \propto -0.5\varphi$, e.g. doubling the number of pixels of the final image will double the CPU time, whereas reducing the field-of-view (φ , typically set at 30 degrees) to halve the value will increase the CPU time by 50%. To measure the performance, we re-calculated with the latest version of the code some individual images of movies presented later in this work. As reference system, we executed Splotch

⁹ <http://openmp.org/>.

¹⁰ <http://www.gnu.org/licenses/gpl.html>.

as single task on an AMD Opteron 850 CPU with 2400 MHz. The underlying cosmological simulation for the movie described in section 4 is described by ≈ 11 million gas particles and ≈ 5 million star particles, of which a large fraction is usually visible in the individual frames. On our reference system Splotch takes ≈ 67 s to render a typical frame (as shown in figure 7) with a resolution of 1400×1050 pixels. The *zoomed* simulation used for making the movie presented in section 3.2 is represented by a slightly smaller number of gas particles (in this case ≈ 5 million), but focuses on a smaller region in space revealing more details. To render a typical frame with a resolution of 800×800 pixels Splotch takes ≈ 72 s on our reference system. Note that to produce the movies a substantial fraction of additional time is needed for interpolating the particle files for the individual frames.

3. Application to cosmological simulations

Clusters of galaxies are ideal cosmological probes (see Borgani and Guzzo (2001), for a review). They are the largest collapsed objects in the Universe and so are very sensitive to the structure-formation process. The evolution of the diffuse cluster baryons (the so-called intra-cluster medium, ICM) has a nontrivial interplay with the cosmological environment, as well as with the processes of star formation and evolution of the galaxy population. Modern cosmological, hydro-dynamical simulations are able to model a large variety of physical processes, among them also the energy released by stellar populations and nuclear activity in galaxies and clusters of galaxies, the largest known structures the universe is built of. This energy changes the thermodynamical properties of the gas, generates complex patterns in the ICM, so as to regulate the process of gas cooling and star formation. At the same time, the infall of smaller structures within the main cluster regions continuously perturbs the conditions of dynamical equilibrium. This is evident both from the dynamics traced by galaxy motions within and around clusters, and from the presence of prominent features within the ICM (e.g. cold fronts and bow shocks) which represents the signature of merging structures. One of the most common publically available cosmological simulation codes is P-GADGET2 (Springel 2005). Highly optimized, it is used for performing some of the largest cosmological simulations available at the moment, including various physical processes at work during the evolution of cosmic structures. The visual representation of the data produced by such simulations can provide substantial help to understand the structure and the dynamics of the matter filling our universe. In the following two subsections, we show some examples extracted from scientific work presented in astrophysical journals or conferences. In all the cases, the visualization was based on Splotch.

3.1. Visualizations using flat projection

For visualizing cosmological structures, it is often useful to adopt a flat projection to obtain the equivalent of a map as obtained from a very distant object. In this case, the ray-tracing—due to handling of absorption—still can lead to some 3D impression, as becomes apparent in figure 1. Especially in the temperature map (right panel), the interplay between the hot atmosphere (red) and the cold filaments (white), which squeeze like pillars into the atmosphere, is particularly emphasized by the ray-tracing technique used.

Although the ray-tracing algorithm is motivated in the previous section to obtain a realistic, 3D representation, sometimes it can be useful to modify it slightly (in an unphysical manner) to obtain interesting diagnostics of cosmological simulations. Figure 2 shows such an

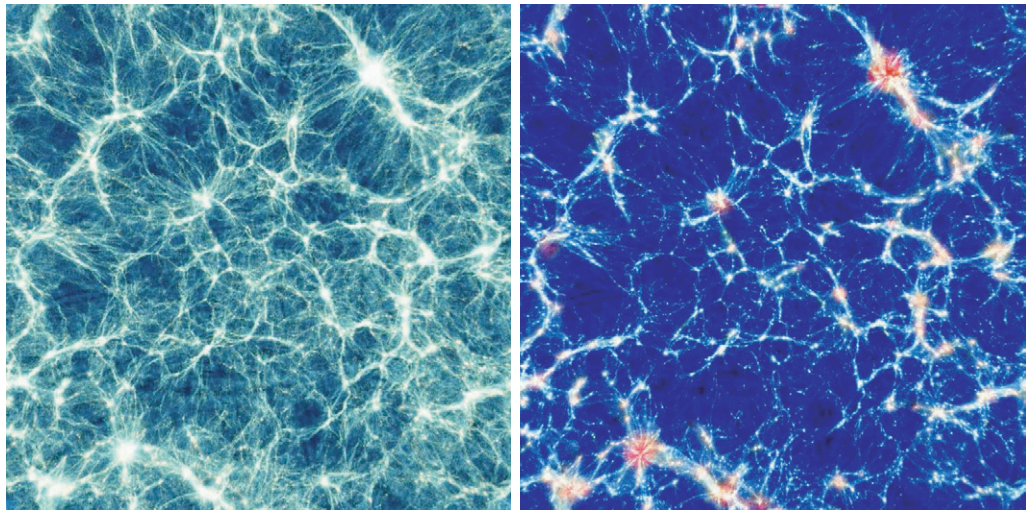


Figure 1. Map of a cosmological simulation box. The light would need nearly 10^9 years to cross the sidelength of the box and the slice thickness corresponds to ≈ 50 million lightyears¹¹. The visualization has used a flat projection (i.e. disabling perspective). The largest structure formed in this cosmological simulation (a massive galaxy cluster) can be found in the upper right side of the panel. Left panel colours the density of individual particles, right panel colours the temperature of individual particles. Taken from Borgani *et al* (2004).

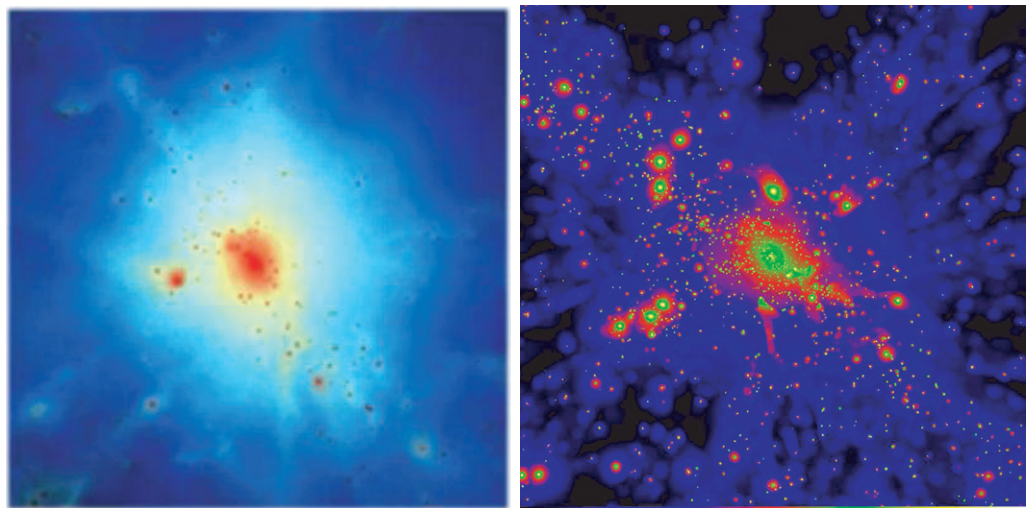


Figure 2. Map of two galaxy clusters obtained by using a flat projection, but associating the displayed value in inverse order as ‘depth’, i.e. ray-tracing from low values to high values. The substructure present in the hot atmosphere of the clusters can be easily recognized. Left panel is taken from Borgani *et al* (2004) and the right panel is taken from Borgani *et al* (2006).

¹¹ This corresponds to ≈ 274 and ≈ 17 Mega-parsec (Mpc) respectively. The visualization is using a flat projection (i.e. disabling perspective).

example. To emphasize the presence of small-scale substructure, which in the real ray-tracing would be partially (or even fully, if positioned far inside the dense atmosphere) absorbed by the outer parts of the atmosphere, the sorting of the particles is not done according to their spatial position along the z -axis but as a function of relevant scalar quantities associated with each particle (e.g. temperature, metallicity—left panel) or of their density (right panel). Therefore, the large amount of small structures (in this case galaxies) and their surrounding becomes clearly visible, as they are passed by the ray only after the rest of the atmosphere is passed. In this way, the 3D impression is obviously lost. Although this kind of map could be obtained by other adaptive map-making algorithms, it is worth mentioning that in *Splotch* this different behaviour is basically obtained by changing only one parameter which determines the desired sorting technique.

3.2. Animations of evolving galaxy clusters

To display the dynamics characterizing the formation of galaxy clusters, the realistic, 3D impression obtained when rendering with *Splotch* results in amazing animations, as shown in figure 3. This movie is based on processing nearly 250 outputs (each ≈ 1 GB in size) of a high-resolution, ‘zoomed’ galaxy cluster simulation taken from Borgani *et al* (2006), who followed with the camera a slightly elliptical orbit around the forming object. To obtain the high number of evolving frames, the individual snapshots were interpolated 20 times in a pre-processing step. Therefore, a total amount of ≈ 5 TB of temporary particle data were processed to show the evolution of a galaxy cluster from very early times on. The representation starts when the universe was just 5% of its present age, and the first galaxies were taking shape (around $z \approx 6$)¹². Light would need about 30 million years to pass the region of space¹³ shown in this animation. The images show the intergalactic medium coloured by its temperature. At z around 3.5, the universe was 15% of its current age, and the forming large-scale structure (filaments) can be clearly recognized. The inlay in the lower right shows a zoom into the interior of one of the two prominent proto-clusters. This is the result from a second run of *Splotch* with different settings and was superimposed on the original frame in a post-processing step. In such structures (clusters of galaxies), several thousands of galaxies can be bound by gravity. At z around 0.8, the universe is half as old as it is now, and the two prominent proto-clusters begin to merge into one galaxy cluster. Such events are the most energetic phenomena since the universe was born in the Big Bang. In the final phase of this merging event, a gigantic shock-wave is initiated, releasing enormous amounts of energy. The shock can be identified as a red, outward-moving shell. The movie can be downloaded from http://www.mpa-garching.mpg.de/galform/data_vis/index.shtml#movie8, where a number of other movies of evolving galaxy clusters produced with *Splotch* can be found.

In some cases, even a flight through a non-evolving representation of cosmological structures can be very enlightening. To optimize such a case, *Splotch* allows us to read the geometrical setup (i.e. the camera position \vec{c} , the look-at point \vec{l} and the sky vector \vec{s}) from a plain ASCII file. Thereby, it produces one frame for every entry in the file, without having

¹² In cosmology, instead of quoting the past time (in unimaginable units of Giga-years) the so-called redshift z is often used, as it is a direct observable and measures the expansion of the universe between the time when the photon was emitted and the present.

¹³ This corresponds to $\approx 3 \times 10^{25}$ cm ≈ 10 Mega-parsec (Mpc).

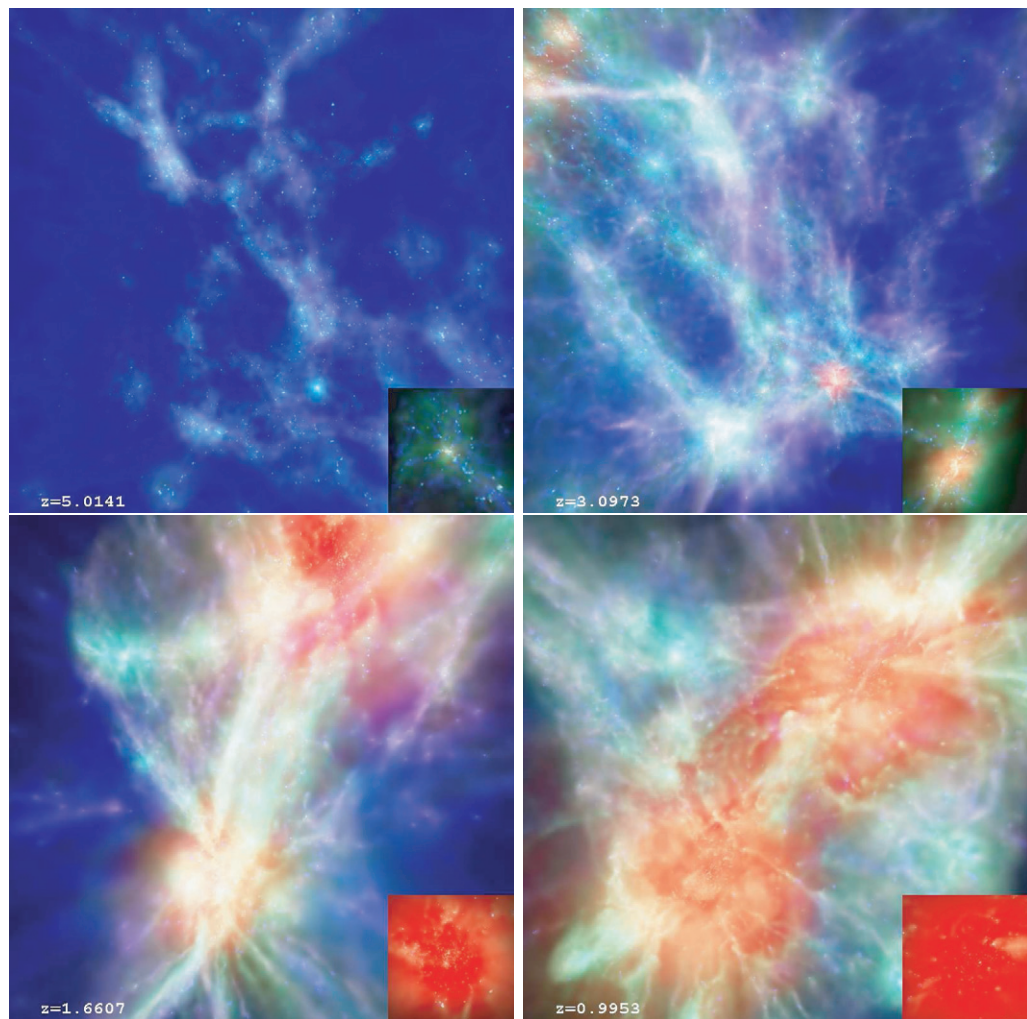


Figure 3. Sequence of ray-tracing the formation and evolution of a galaxy cluster from a high-resolution ‘zoomed’ simulation (Borgani *et al* 2006). Here, the ray-tracing approach by its absorption feature creates impressive features similar to moving through fog when the camera passes through small objects on its way around the structure. The upper panels show a region of the universe at an early time. It is rich in filamentary structures, and later, when the universe evolved, almost all these structures collapsed to form a single, prominent galaxy cluster. The lower panels show the same region at a later time, where the two main progenitors of the final galaxy cluster merged. The movie can be downloaded from http://www.mpa-garching.mpg.de/galform/data_vis/index.shtml#movie8.

to reload the cosmological simulation every time. This can be quite handy, especially when the simulation data are huge (several GB or more). It is also useful to pre-process the data to emphasize different aspects of a cosmological simulation. For example, figure 4 shows a flight through an extremely highly resolved galaxy cluster taken from Dolag *et al* (in preparation, more than 25 million particles within the cluster). Here, different components have been separated in a post-processing step using SUBFIND (Springel *et al* 2004) to separate diffuse and

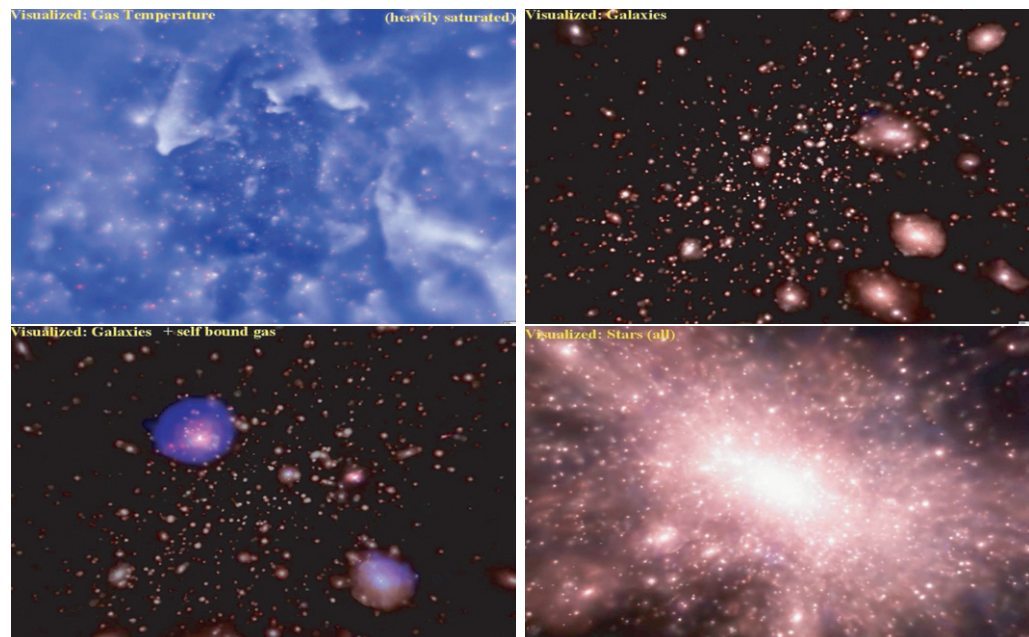


Figure 4. Sequence of ray-tracing of the internal structure of a galaxy cluster (Dolag *et al* in preparation). Here, different components, such as diffuse gas, stars in individual member galaxies, gas bound in such galaxies and diffuse stars within the galaxy cluster are used in different combinations for producing the images. Such visualizations can give unique impressions of the structure and the dynamical interplay between these different components as predicted in such high-resolution ‘zoomed’ simulations. Here, a special colour scheme has been used, to produce colours similar to those in the well-known image of the Orion nebula, obtained by the Hubble Space Telescope. The movie can be downloaded from http://www.mpa-garching.mpg.de/galform/data_vis/index.shtml#movie9.

self-bound substructures within the galaxy cluster. The movie starts from showing the structures in and around the hot atmosphere of a galaxy cluster. After zooming into the cluster, the camera path follows an elliptical orbit around the centre. Prominent structure within the hot plasma becomes visible. Changes of the saturation and colour settings reveal different structures at different scales and their interactions. Some of the substructures inside the cluster are able to maintain a self-bound atmosphere for a while (shown in light blue). The population of free-floating stars, which originates from destroyed galaxies, shows prominent stripes as imprints of the orbits of the former galaxies they belonged to. Despite such destruction, more than one thousand individual galaxies can still be identified within the cluster, forming new stars in their centres (shown in dark blue). Only a small number of these are still maintaining a hot, self-bound atmosphere (shown in light blue). In the zoom-out, all the stars formed within the simulation are shown. The text and some visual blending effects are produced by a post-processing step. This animation can also be downloaded from http://www.mpa-garching.mpg.de/galform/data_vis/index.shtml#movie9.

4. The 4D universe

The visual rendering of the universe and its evolution can represent a spectacular approach towards the intuitive explanation of the complex phenomena that are the bases of our incredibly fascinating theory of cosmology, for both scientists and the general public. Unfortunately, an effective implementation of this approach is extremely challenging, and it often leads to scientifically nonrigorous or even misleading realizations. Even the most rigorous approaches lead to incomplete or partial results, due to the difficulties in effectively giving a graphic representation of the 3D space in which matter, stars and galaxies are born, move, evolve and interact leading to the wonderful pattern that we can observe in the sky.

We have exploited an unprecedented combination of numerical modelling, astronomical observational data and Splotch-based rendering for creating a 4D reconstruction of the universe, as described by the commonly accepted model. The full 3D volumetric representation is provided by the adoption of the stereographic rendering approach, according to the technique described below.

Even more challenging is the fourth dimension, time, which is obtained by describing the dynamics and the behaviour of cosmological structures from their birth to the present epoch, using the results of state-of-the-art computer simulations.

The final outcome is an animation displaying a flight through a volume representative of the universe, following its evolution. In the first part of the animation, the focus is on the complex architecture of filaments and sheets which define the cosmological large-scale structure. We then concentrate on a cluster of galaxies, and thousands of small objects, the galaxies which compose it. Finally, we zoom in, showing the fascinating structures of a spiral galaxy, as we believe our own Milky Way is like. The result of this effort will be displayed in the virtual reality facility of the brand new Turin Planetarium.

In the rest of this section, we will describe the main steps in the realization of the *4D Universe* animation.

4.1. Splotch usage and stereographic rendering

The realization of such a complex visualization project required the usage of Splotch to be properly fine-tuned. Colours, transparencies, smoothing lengths adapt to the represented environment and objects, which continuously change following the evolution of cosmological structures and the movements of the camera.

The movie supports stereographic visualization. Therefore, two different realizations of the movie, one for each eye, must be produced. This is accomplished by generating two different camera paths and creating the frame sequence according to each path. The two paths are calculated as follows.

As described before, the geometry for the rendering is given by three vectors: the camera position \vec{c} , the look-at point \vec{l} and the sky vector \vec{s} . The first two vectors define the viewing vector $\vec{v} = \vec{l} - \vec{c}$. In general, the sky vector \vec{s} does not have to be orthogonal to the view vector \vec{l} to be used in Splotch. However, a simple projection

$$\vec{S} = \vec{s} - \cos(\alpha) \vec{v} \frac{|\vec{s}|}{|\vec{v}|}, \quad (4)$$

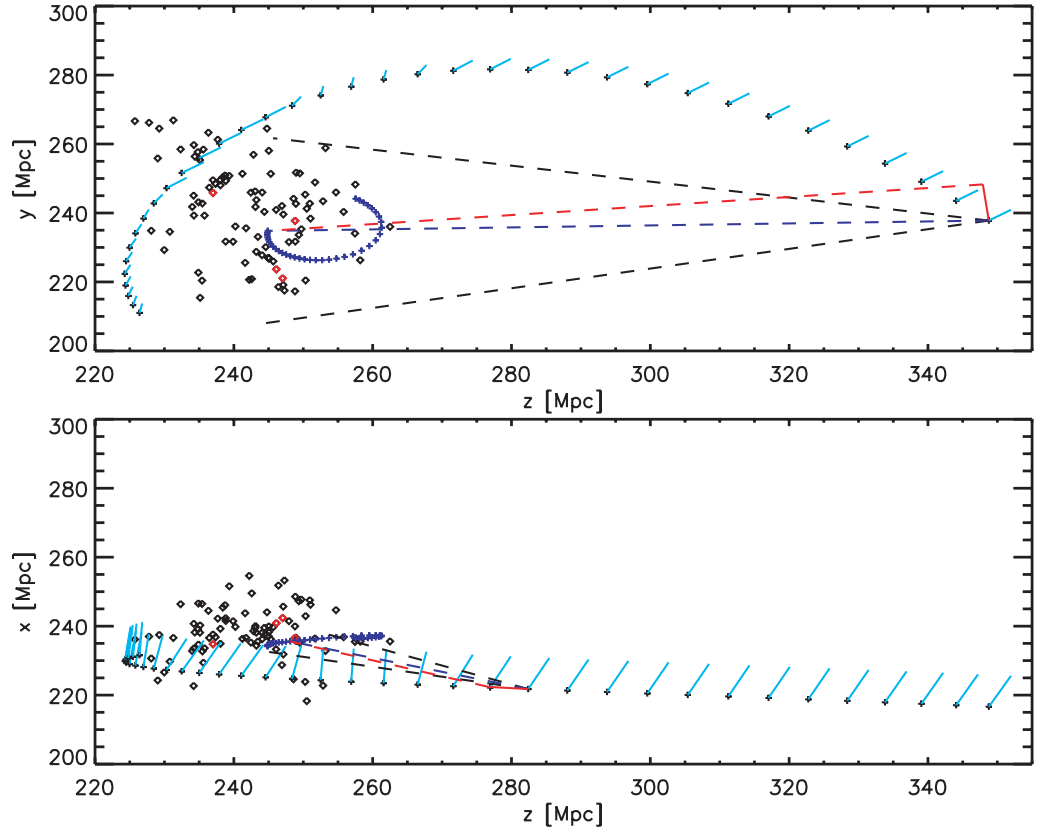


Figure 5. The geometrical setup of the camera path for the cosmological movie in two projections to describe the 3D setup. The diamonds mark the positions of the largest cosmological objects in the simulation volume, the black plus symbols mark the position of the camera with the sky vector \vec{s} attached in light blue. The dark blue plus symbols show the look-at position \vec{l} . Both are shown tracing their movement for the full movie sequence, plotting only every hundredth position. For the first frame, the view vector \vec{v} is shown as blue-dashed line, encompassed by the black-dashed lines marking the visible region defined by the field-of-view angle φ . The red line marks the *right* vector \vec{r} and the red-dashed vector marks the view vector for the right eye. Note that here a (unphysically large) value of $\beta = 6$ was used for visualization purposes.

can be used to obtain its orthogonal part \vec{S} , where α is the angle between the sky vector \vec{s} and the view vector \vec{v} , which can be obtained by the classical formula

$$\cos(\alpha) = \frac{\vec{v} \cdot \vec{s}}{|\vec{v}| |\vec{s}|}. \quad (5)$$

For stereoscopic projection, the same scene is re-rendered from the position of the second eye, calculated as follows. If we parameterize the position of the right eye by a separation angle β , the new camera position for the right eye \vec{C} can be obtained easily as

$$\vec{C} = \vec{c} + \frac{\vec{r}}{|\vec{r}|} \frac{\beta 2\pi}{360} |\vec{v}|, \quad (6)$$

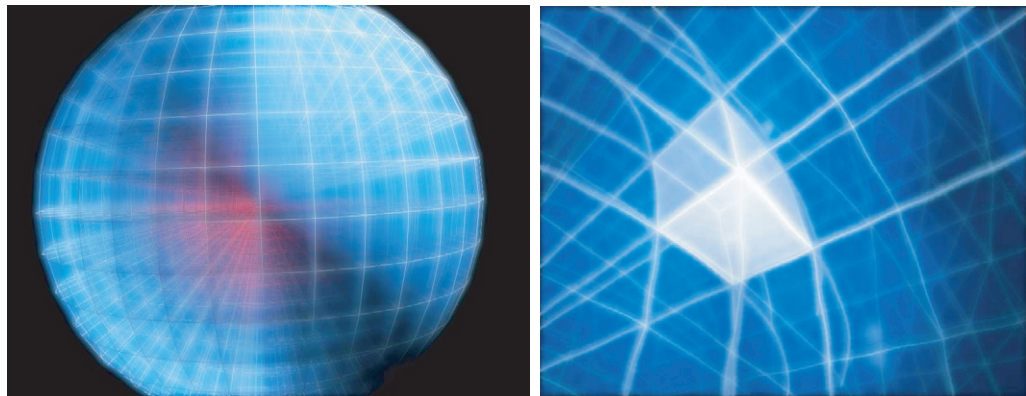


Figure 6. Representation of the geometry of the space–time in the very first instants of the Big Bang.

whereas the vector \vec{r} can be simply obtained by the cross product $\vec{r} = \vec{S} \times \vec{v}$ defining the *right* direction. Later, the two images for the different eye positions can be combined to be used in stereoscopic devices. For the movie, we used values between 0.15 and 0.2 for β , depending on the final device used. Figure 5 shows the geometrical setup for the cosmological part of the 4D universe movie. It shows the relation between the different vectors involved as well.

4.2. The Big Bang

The representation of the first few instants after the Big Bang is an incredibly difficult task. Nothing that we can conceive is able to properly describe what happened in that initial time frame. We have adopted an abstract geometric representation, in which a spherical symmetric mesh represents the expanding space–time (see figure 6). One of the cells of this mesh is selected to represent the sample volume where the subsequent stages, based on the simulated data, take place.

Since Splotch renders only point-like data, this part of the movie, based on polygonal data, has been modelled and rendered using a commercial 3D graphic package: Autodesk 3DStudio Max.

4.3. The evolution of the universe

For a project like the *4D Universe*, it is necessary to create the images making use of data from an extremely high-resolution simulation. We used a simulation of a large cosmic structure which connects four very massive galaxy clusters (see Dolag *et al* 2006). The high-resolution region in this simulation is more than one order of magnitude larger than that of the simulations of individual clusters presented in section 3. Figure 7 shows some frames extracted from the cosmological part of the movie. The flight through an evolving cosmic structure starts at early times (the so-called dark ages), where the matter in the universe is in a cold and neutral state. First, objects collapse and form the first proto-galaxies embedded in a heated atmosphere, hosting the first stars and quasars (QSOs). The high-energy light from those stars/QSOs heats and ionizes the material in the universe. Although cosmological simulations cannot take this effect into account self-consistently, the adoption of a time-dependent background radiation

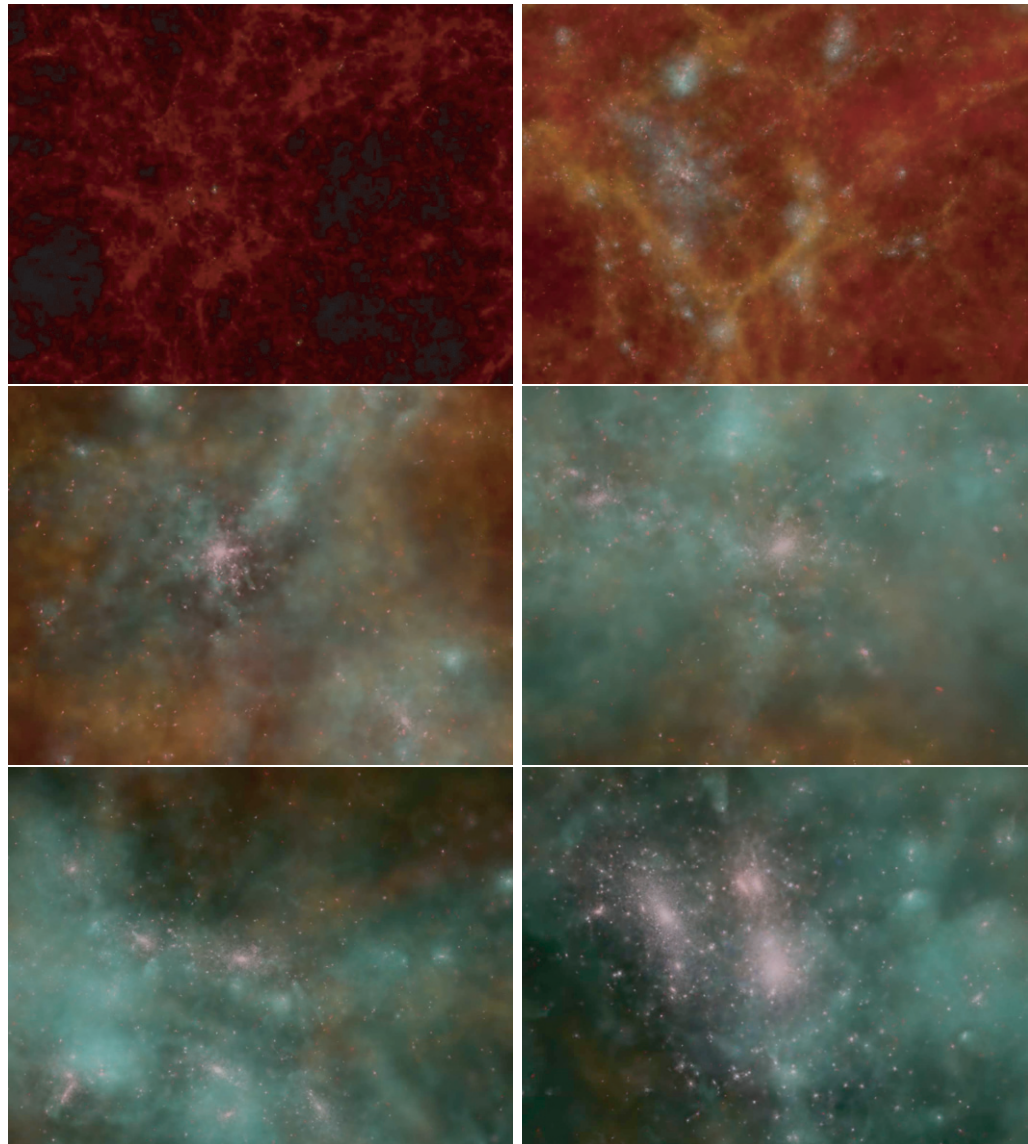


Figure 7. Sequence of ray-tracing a flight through the evolving structure from a cosmological simulation. Here, different components, gas and stars, are visualized together, but using two different colour tables. The colour table for the gas is specially trimmed to become transparent at high densities, so that stars inside the forming gaseous atmosphere are visible. A compressed version of the *4D Universe* movie can be obtained from http://www.mpa-garching.mpg.de/galform/data_vis/index.shtml#movie11.

field mimics the effects of re-heating the universe and reveals the fine, filamentary structures formed until that point. Larger and larger structures form within a long and violent process of merging of smaller structures leading to the formation of the galaxy clusters, still connected to each other by filamentary structures. At the end, the movie zooms into a spiral galaxy, similar to what we expect the Milky Way to look like, for a final fly-by. In this case, the galaxy is artificially constructed from an astronomical image (as described in the next section), as it is still impossible to obtain such detailed and realistic galaxies directly within cosmological simulations.

4.4. The galaxy

Galaxies, and in particular spiral galaxies, are amongst the most spectacular objects in the universe. For many galaxies, high-resolution images have been caught (e.g. by the Hubble Space Telescope, <http://heritage.stsci.edu/2005/12a/big.html>), which provide a detailed description of their structure and morphological features.

However, for the sake of 4D visualization, several difficulties must be overcome. Firstly, even the ultimate cosmological simulations fail to produce spiral galaxies as detailed as we observe them in the real world. This is due to actual limitations in resolution and lack of possibility of properly including all key physical processes important to form a spiral galaxy from first principles. Secondly, any observation of a galaxy reflects only 2D, projected images of the object. The detailed 3D shape of the galaxy is unknown. Furthermore, no time evolution can be observed, the dynamical timescale being hundreds of millions of years. We solved these problems by a proper modelling of the galaxy and its evolution, based on observations combined with theoretical predictions of their dynamics.

The galaxy model is built in two main steps: the reconstruction of the 3D geometry (based on the *Galaxian* code) and the time evolution (based on the *EvolveGalaxy* code).

Step 1: the *Galaxian* code

In order to reconstruct the spatial distribution of the basic matter components of the galaxy (the disk, with stars and gas, the bulge and the globular cluster) we have developed a code called *Galaxian*, which works as follows.

- A specific galaxy is selected and its high-resolution image converted to a raw format (more easily readable by our code) with a depth of 256 colours. This is the main data input of the code. For the planetarium movie, we have chosen the M51a spiral galaxy, also known as the *Whirlpool Galaxy*.
- In order to generate a 3D distribution close to that of the gas distributed in the galaxy, we define for each image's pixel a point, with z -coordinate randomly extracted from a Gaussian distribution centred on the galactic plane, setting σ_z at much smaller radius than the galaxy. In this way, we imitate the planar geometry of the spiral galaxy. Around each point, a cloud of N points is generated with spherical symmetry. The points get the colour of the pixel in which they fall. This permits us to define a volumetric point distribution, avoiding, at the same time, getting a *pixelled* aspect: points uniformly distributed along z on the same pixel having the same colour. This would lead to the emergence of unphysical monochromatic columns.
- Disk stars are represented by points with random positions covering the whole galaxy map. The three coordinates are generated following a Gaussian probability function with $\sigma_x \sim \sigma_y \gg \sigma_z$ in order to get an oblate spheroidal distribution. Bulge stars are generated at the centre of the galaxy with a quasi-spherical distribution.
- Finally, a spherically symmetric distribution of globular clusters is generated around the galaxy. Each globular cluster accounts for several thousand stars (the precise number being random).

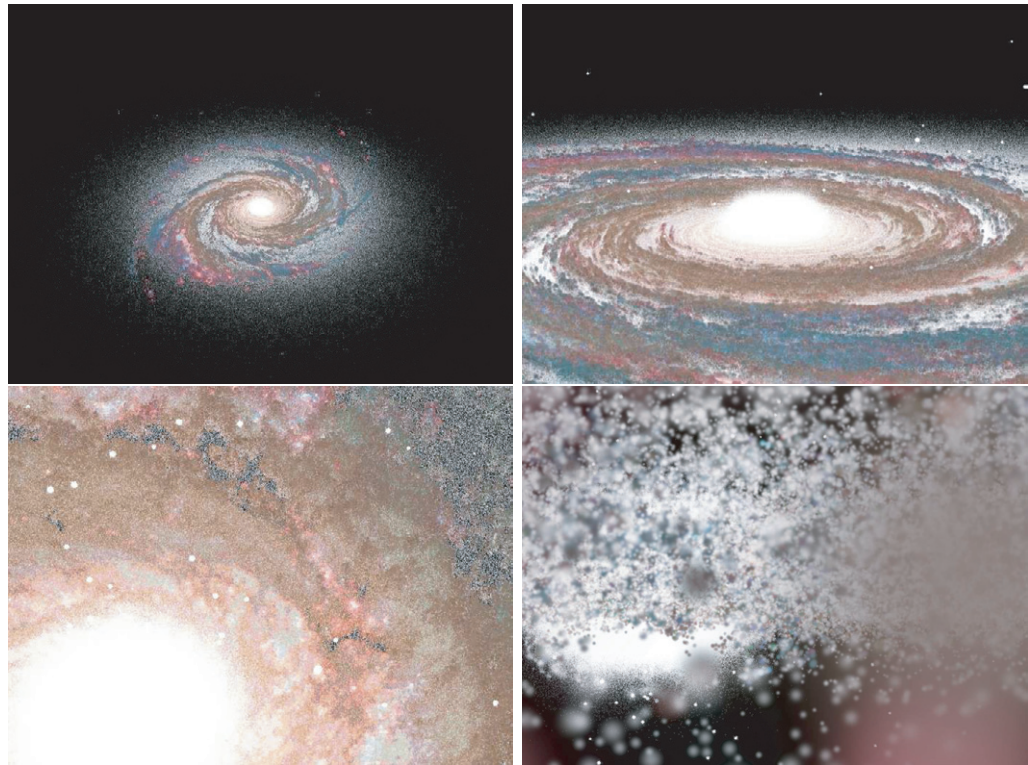


Figure 8. Sequence of ray-tracing a flight through a reconstructed galaxy planted into the cosmological simulation. Here, the different components, gas in the disk, stars in the disk, bulge and globular clusters are visualized together. However, every gas and star particle has a vector array directly reflecting its rgb colours, which is used instead of the colour look-up table. The reconstruction is based on the colours composed from a real, optical observation (see text for details). The particles of the galaxy are added to the cosmological simulation and ray-traced in one shot, so that consistent movement of the galaxy and the background is accomplished. A compressed version of the *4D Universe* movie can be obtained from http://www.mpa-garching.mpg.de/galform/data_vis/index.shtml#movie11.

Step 2: the *EvolveGalaxy* code

Giving a self-consistent numerical description of the galaxy dynamics is an outstanding task, which is only marginally accomplished by the most advanced and expensive numerical simulations. For our purposes, a simpler approach is sufficient. Such an approach is implemented in the *EvolveGalaxy* code, according to the following steps.

- A typical galaxy rotation curve is used for making all the gas and star points on the disk rotate, according to the differential rotation law. This results in a purely kinematic approach (no dynamics considered).
- The stars in the bulge and in the globular clusters rotate around the galaxy centre, each with their own constant velocity, calculated in order to balance the gravitational attraction of the matter distributed inside the orbit.
- Equations of motion at the disk points are integrated using a simple first-order integration method. Due to the kinematic approach and the amplification of numerical errors, the

description is acceptable for a limited time range only. After a few hundred million years, the point distribution becomes meaningless and the simulation must be stopped.

- At proper time intervals, the point distribution can be rendered calling the *Splotch* function directly from inside the *EvolveGalaxy*. The code can be used on multiprocessor computers, making each computing element calculate a different fraction of the whole time evolution, strongly accelerating the rendering process. Figure 8 shows the result of this process. A compressed version of the *4D Universe* movie can be obtained from http://www.mpa-garching.mpg.de/galform/data_vis/index.shtml#movie11. Note that in this version, the galaxy was implanted directly within the cosmological simulation and both the large-scale structure as well as the galaxy were rendered at once. Here, all geometrical calculations as well as the position data were adapted to have double precision to handle the large dynamical range between the galaxy and the cosmological structure. Therefore, the parts zooming into and out from the galaxy, as well as the background structure when flying around the galaxy appear self-consistent within the visualization.

5. Conclusions

Visualization is one of the most effective techniques for exploring and presenting a scientific dataset, understanding at a glance its basic features and properties. Proper graphical solutions must be developed and exploited in order to fully achieve such objectives. Such solutions can change according to the target users, depending, for example, on whether professionals or the general public are being addressed. Moreover, different application fields can have specific requirements and different datasets can have peculiarities which need to be treated with a suitable approach.

Furthermore, the software should be able to face the major challenge represented by present time scientific datasets: the huge data volume. Visualization tools have to be able to manipulate gigabytes of data at once, producing the result in an acceptable time. The algorithm must be tuned to the data in order to get the best possible performance. 64bit architectures, large memories and multiprocessor systems have to be supported and exploited in order to handle such extraordinary data-processing efforts.

In this paper, we have shown how the *Splotch* software can be adapted to a variety of different applications in astronomy, ranging from professional usage to divulgence and outreach content production. For point-like datasets, *Splotch* meets all the previous requirements, adding the further advantage of being open source, flexible and easily extensible. It can be used both as a stand-alone application and as a library function, callable from inside the simulation code.

Splotch was successfully used for producing a stereographic movie for the Turin Planetarium. The movie has the ambitious goal of describing the birth and the evolution of the universe, exploiting actual cosmological numerical simulations, observations and models. Therefore, it helps to give the general public an exceptional insight into our understanding of the world on its largest scales.

As a side product of the increased complexity of the physical systems captured by state-of-the-art cosmological simulations, visualizations of them are extremely attractive for the public and thus start reaching a state where they are of comparable beauty to real observations, which traditionally reflect the extremely attractive nature of astronomy.

Acknowledgments

KD acknowledges financial support from the ‘HPC-Europa Transnational Access program’ and the hospitality of CINECA, where part of the work for the *4D Universe* was carried out.

References

- Borgani S, Dolag K, Murante G, Cheng L M, Springel V, Diaferio A, Moscardini L, Tormen G, Tornatore L and Tozzi P 2006 *Mon. Not. R. Astron. Soc.* **367** 1641–54
- Borgani S and Guzzo L 2001 *Nature* **409** 39–45
- Borgani S, Murante G, Springel V, Diaferio A, Dolag K, Moscardini L, Tormen G, Tornatore L and Tozzi P 2004 *Mon. Not. R. Astron. Soc.* **348** 1078–96
- Dolag K, Borgani S, Schindler S, Diaferio A and Bykov A M 2008 *Space Sci. Rev.* **134** 229–68
- Dolag K, Meneghetti M, Moscardini L, Rasia E and Bonaldi A 2006 *Mon. Not. R. Astron. Soc.* **370** 656–72
- Dolag K *et al* in preparation
- Gingold R A and Monaghan J J 1977 *Mon. Not. R. Astron. Soc.* **181** 375–89
- Lucy L B 1977 *Astrophys. J.* **82** 1013–24
- Monaghan J J and Lattanzio J C 1985 *Astron. Astrophys.* **149** 135–43
- Shu F 1991 *Physics of Astrophysics: Volume I Radiation* (New York: University Science Books)
- Springel V 2005 *Mon. Not. R. Astron. Soc.* **364** 1105–34
- Springel V, White S D M, Tormen G and Kauffmann G 2001 *Mon. Not. R. Astron. Soc.* **328** 726–50