Interactive Cosmology Visualization Using the HUDF

Matt R. Mechtley[1], Rogier A. Windhorst[1], Seth H. Cohen[1], Lisa M. Will[2]

ABSTRACT

We have developed a Java[3]-based teaching tool, “Appreciating Hubble at Hyperspeed” (“AHaH”), intended for use by students and instructors in beginning astronomy and cosmology courses, which we have distributed via the World Wide Web[4]. This tool lets the user hypothetically traverse the Hubble Ultra Deep Field in three dimensions at over $\sim 500\times10^{12}$ times the speed of light, from redshifts $z=0$ to $z=6$. Users may also view the Universe in various cosmology configurations and two different geometry modes – standard geometry that includes expansion of the Universe, and a static pseudo-Euclidean geometry for comparison. In this paper we detail the mathematical formulae underlying the functions of this Java application, and provide justification for the use of these particular formulae. These include the manner in which angular sizes of objects are calculated in various cosmologies, as well as how the application’s coordinate system is defined. We also briefly discuss the methods used to select and prepare the images in the application, the data used to measure the redshifts of the galaxies, and the qualitative implications of the visualization – that is, what exactly users see when they “move” the virtual telescope through the simulation.

Subject headings: Data Analysis and Techniques

1. Introduction

In beginning astronomy courses, many non-science majors appear to have a significant lack of understanding – even after taking the introductory courses – of basic concepts such as wavelength, electro-magnetic spectrum, speed-of-light, look-back time, redshift, and expansion of the Universe. We believe this lack of concept acquisition or retention represents a significant shortcoming of the currently available teaching tools. While pictures, figures, and other static media are certainly effective at communicating many concepts, they tend to be poor at showing effects in three dimensions or that evolve over time. Since virtually all cosmological effects require very large time or distance scales to become apparent, a different teaching medium is preferable.

---

[1] School of Earth and Space Exploration, Arizona State University, Tempe, AZ 85287

[2] San Diego City College, San Diego, CA 92101

[3] http://java.sun.com

[4] http://www.asu.edu/clas/hst/www/ahah/download.html

“Appreciating Hubble at Hyper-speed” is an educational tool that aims to address these issues of concept acquisition and retention by providing a visual and interactive learning medium. The project uses data from the HST Cycle 12 Project “GRAPES” (Grism-ACS Program for Extragalactic Science; Pirzkal et al. 2004) to build a redshift-sorted database of over 5000 galaxies within the Hubble Ultra Deep Field (HUDF). These galaxies range from redshift $z \approx 0.05$ to $z \approx 6$, spanning nearly 90% of the history of the Universe (Yan & Windhorst 2004; Bouwens et al. 2006). Since these data represent the deepest optical image of the Universe ever obtained, they are thus uniquely suited to students understand the effects of the expanding Universe.

2. Data Selection and Preparation

We first created a custom-balanced RGB version of the HUDF image. While the image provided in the original press releases would have been adequate, it has the undesirable characteristic that very bright areas, such as bulges in large spirals, appear burnt-out and lack fine detail. The raw HUDF data consist of B-, V-, i'-, and z'-bands (Beckwith et al. 2006), so we created a three-channel color image by first combining the B- and V-bands, applying weights based on the sky SNR[5]. We then used the algorithm developed by Lupton et al. (2004) to create the combined RGB image, with the combined $B{+}V$-bands as the blue channel, the i'-band as the green channel, and the z'-band as the red channel[6]. Besides showing more detail in bright areas, this method has the added benefit that an object with a specified astronomical color has a unique color in the composite RGB image. A comparison of the original STScI color images and our prepared images is shown in Figure 1. The full HUDF image using this color preparation technique is also available as an interactive map on the World Wide Web[7].

The galaxies represented in the AHaH application were i'-band selected using `SExtractor` with a detection threshold of $\sigma = 1.8$ above sky. The i'-band dropouts of Yan & Windhorst (2004) were later added by hand. We then created color JPEG “stamp” images for each individual object, using the `SExtractor`-generated segmentation map to mask as black any pixels outside the detected source. These “stamps” were then converted pixel-for-pixel to PNG images, which employ a lossless compression algorithm – no image quality was thus lost. We then developed a transparency map based on each pixel’s brightness, which was saved into the PNG alpha channel. The resulting images can thus be displayed as semi-transparent, allowing objects in the distance to show through the dim regions of objects in the foreground.

Photometric redshifts for the galaxies were measured with `HyperZ` (Bolzonella, Miralles, & Pelló

---

[5] 0.765 weight in V and 0.235 weight in B

[6] The channels were first scaled as follows, proportional to the data zero points – Red: 716.474, Green: 345.462, Blue: 254.449

[7] http://wwwgrapes.dyndns.org/udf_map/udfmain_low.html

2000), using a combination of the original HST-ACS four-band ($BVi'z'$) data from the HUDF, along with J- and H-band data from HST-NICMOS (Thompson et al. 2005). We have supplemented the photometric redshifts with spectro-photometric redshifts measured by Ryan et al. (2007), which incorporate the aforementioned $BVi'z'JH$ data as well as grism spectra from GRAPES (Pirzkal et al. 2004), U-band observations from CTIO-MOSAIC II, and K_s-band data from VLT-ISAAC. For a summary of all these data and the quality of the spectro-photometric redshifts, see Ryan et al. (2007). When available we have chosen to use the more reliable spectro-photometric redshifts.

3. Development of Formulae

As previously discussed by Wright (2006), there are a number of different methods for calculating distances in cosmology. For our purposes, the most meaningful of these is the comoving radial distance, D_R, representing the spatial separation of an object and an observer with zero peculiar velocity at a common time. This distance takes into account the expansion of the Universe, and so is more useful when dealing with distances on very large scales (and thus very large look-back times), as is the case with galaxies in the HUDF. Henceforth, we shall adopt the convention of referring to the comoving radial distance from Earth to a galaxy as D_R, and the comoving coordinate distance between two arbitrary points in the coordinate system as r_{ij}.

We also wish to calculate the angular sizes of objects as they would be observed from redshifts other than zero. To do so, we need a formula for the angular size distance, D_A. That is, the distance which satisfies the equation $d=\theta D_A$ for an object with transverse diameter d subtending an angle θ in the field of view. In a simple Euclidean space, this is the same as the radial distance, but again we must take into account the expansion (and possible curvature) of the Universe, so we must use a separate equation in the AHaH tool.

Additionally, we need to consider how we wish to define the coordinate system for the objects within the Java tool. Although we have very deep HST imaging data that allow us to show how the Universe has changed over time, all of these data were collected at a common time (2003/2004). Moreover, the principal distance measure that we have available, D_R, also assumes a common time. Thus the most sensible coordinate system is one with three spatial dimensions that makes all calculations for a common time, viz. when the data were collected. We can then contract the distances in this "comoving coordinate system" as necessary to simulate observations from redshifts other than zero. The question remains of how we should derive such coordinates from the data that we have in such a way that they will be useful to us – this is discussed in § 3.3 below, prior to deriving the equations.

3.1. Comoving Radial Distance

To begin, we need the comoving radial distance, D_R, from the Earth to an object at redshift z, derived from the Robertson-Walker metric, as discussed previously, e.g., Longair (1998, Ch. 7), Ryden (2003, Eq. 6.8), and Wright (2006, Eq. 6). We express this as the integral:

$$D_R = \int \frac{c \cdot dt}{a} = \int_{\frac{1}{1+z}}^{1} \frac{c \cdot da}{a\dot{a}} = \frac{c}{H_0} \int_0^z \frac{dz}{(1+z)\dot{a}}, \tag{1}$$

where the scale factor $a = 1/(1+z)$. The derivative of a with respect to time, $\dot{a}$, is given by the expression:

$$\dot{a} = (\Omega_M/a + \Omega_R/a^2 + \Omega_\Lambda \cdot a^2 + \Omega_K)^{1/2}, \tag{2}$$

where Ω_M, Ω_R, Ω_Λ, and Ω_K are energy density parameters, corresponding to the fractions of the Universe's total average energy density that are attributable to matter, radiation, dark energy, and the curvature of the spatial geometry, respectively. Note that it is assumed these are the only meaningful contributions to the total energy density – that is: $\Omega_M + \Omega_\Lambda + \Omega_R + \Omega_K = 1$.

We evaluate this integral in steps of 0.05 in z from $z = 0$ to $z = 20$ to create a look-up table, interpolating linearly to find the value for any arbitrary redshift. This is because we must make the calculation frequently and for many objects, so computing the integral manually every time would be computationally prohibitive. The resultant error in this method is generally small enough that it translates to less than one pixel's difference even on high-resolution displays, so it can safely be ignored for the purposes of the application. We evaluate the integral using the simple midpoint method, which may not be the optimal solution, but was simple to implement and adequately efficient. As with the linear interpolation, higher accuracy numerical integration would result in less than one pixel's difference when displayed.

3.2. Angular Size Distance

To develop the angular size distance, D_A, we first need the comoving tangential distance, D_T, of an object at redshift z_j as measured by an observer at redshift z_i. This distance is given by the formula:

$$D_T(z_i, z_j) = \begin{cases} \Re' \sin(r'_{ij}/\Re') & \text{if } \Omega_K < 0 \\ r'_{ij} & \text{if } \Omega_K = 0 \\ \Re' \sinh(r'_{ij}/\Re') & \text{if } \Omega_K > 0 \end{cases}, \tag{3}$$

where $\Re'$ is the radius of curvature of the spatial geometry at redshift z_i and r'_{ij} is the value of the comoving coordinate distance at the same redshift (Longair 1998; Wright 2006). These correspond to the cases where the spatial geometry is spherically curved, flat, and hyperbolically curved, respectively. Recalling that both r'_{ij} and $\Re'$ scale as $1/(1+z_i)$, and that $\Re = (c/H_0)/\sqrt{|\Omega_K|}$, we may rewrite D_T as:

$$D_T = \frac{\delta_{ij}}{1+z_i} r_{ij}, \tag{4}$$

where δ_{ij} is simply some function of r_{ij}.

We first define an intermediate quantity U_{ij}[8], representing the argument of sin and sinh in Equation (3).

$$U_{ij} = r'_{ij}/\Re' = r_{ij}/\Re = (H_0/c)\sqrt{|\Omega_K|}r_{ij} \tag{5}$$

Now substituting U_{ij} into Equation (3) above, we get the following expression for δ_{ij}:

$$\delta_{ij} = \begin{cases} \frac{\sin(U_{ij})}{U_{ij}} & \text{if } \Omega_K < 0 \\ 1 & \text{if } \Omega_K = 0 \\ \frac{\sinh(U_{ij})}{U_{ij}} & \text{if } \Omega_K > 0 \end{cases} \tag{6}$$

Note that δ_{ij} expressly depends upon r_{ij}. The case where $\Omega_K = 0$ comes from the limit of both $\sin(U_{ij})/U_{ij}$ and $\sinh(U_{ij})/U_{ij}$ as $\Omega_K \to 0$ – one may observe that Equation (4) then simplifies to $r_{ij}/(1+z_i)$, which is precisely r'_{ij} as in Equation (3).

Thus, using the equation relating the angular size distance and the tangential distance as developed by Longair (1998, Eq. 7.50), the angular size distance from redshift z_i to z_j is given by:

$$D_A = \frac{1+z_i}{1+z_j} D_T = \frac{\delta_{ij}}{1+z_j} r_{ij} \tag{7}$$

3.3. Comoving Coordinate System

Now that we have developed formulae for D_R and D_A, we can consider the best way to create a coordinate system for the Java application. The data we start with are the redshift of an object (with which we can calculate D_R) and four angles: the object's angular size (from the height and width of its image) and the angular separation between the object and the x and y axes, which we define as lines going through the center of the original image. These angles are calculated by taking the corresponding size in pixels and multiplying by the scale in arcsec/pixel of the original HST image[9].

We would like to use this information to create a coordinate system with the original telescope position at the origin. In a Euclidean space this would present no problem, but we have already remarked that the *observed* angles are not the same in an expanding Universe as they would be in a Euclidean space. Further, it would be desirable for the Euclidean coordinate distance to correspond to the comoving radial distance, as this would make calculations significantly simpler. We can accomplish this, but when we create coordinates for each object as such, we need to "correct" the

---

[8] Though defining U_{ij} in this way means we must multiply δ_{ij} by r_{ij} to obtain D_T, it decreases the total number of calculations that we must make within the Java application, since we may calculate U_{ij} once per object and re-use it.

[9] 0".03 per pixel

angles. That is, we want a "Euclidean angular size" associated with a certain observed angular size. We will call this θ_E. An object's angular size is related to its physical transverse diameter, d, by the equation:

$$d = \theta D_A = \theta \frac{\delta_{ij}}{1+z_j} r_{ij} = \theta_E \frac{1}{1+z_i} r_{ij} \tag{8}$$

Note that in the Euclidean case we must contract r_{ij} by a factor of $1/(1+z_i)$ to get the comoving distance from z_i to z_j as measured from z_i (r'_{ij} in Equation (3) above). This is because the proper spatial separation in the current epoch has been stretched by the Universe's expansion, so from redshift z_i it must be scaled appropriately.

Thus canceling r_{ij}, we get the following expression for θ_E:

$$\theta_E = \theta \delta_{ij} \frac{1+z_i}{1+z_j} \tag{9}$$

In our initial data z_i is simply zero, we create coordinates (X, Y, Z) for an object like:

$$X = \sin\left(\frac{\delta_{ij}\theta_X}{1+z}\right) \cos\left(\frac{\delta_{ij}\theta_Y}{1+z}\right) D_R, \tag{10}$$

and similarly for Y and Z. We have thus developed a coordinate system of X, Y, and Z in Mpc with the original telescope position at the origin.

3.4. Simulating Observations From Vantage Points Other Than $z = 0$

Now, when we "move" the camera, we do so by some X_c, Y_c, and Z_c in the coordinate space. By construction, the distance measure here is just the Euclidean coordinate distance:

$$D_E = ((X - X_c)^2 + (Y - Y_c)^2 + (Z - Z_c)^2)^{1/2} \tag{11}$$

Now to determine where to display an object after we have "moved" the camera, we use the distance calculated with Equation (11) and the Euclidean angular size. Using the object's redshift, z_o, and the camera's user-defined redshift, z_c, we rearrange Equation (9) to get:

$$\theta = \theta_E \frac{1+z_o}{\delta_{ij}(1+z_c)} \tag{12}$$

In this case, θ_E is a quantity that we must calculate from our coordinates in the usual Euclidean way.

For an object's angular size it is even simpler than for its (X, Y, Z) position, since we do not have to manually calculate θ_E. We know that in the Euclidean case:

$$d = \theta_0 D_R = \theta_E D_E, \tag{13}$$

where θ_0 is the Euclidean angular size from redshift zero, and D_E is the coordinate distance from the camera to the object. We then solve for θ_E and substitute into Equation (12) to obtain an expression for the desired angular size θ as observed from z_c:

$$\theta = \theta_0 \left(\frac{D_R}{D_E}\right) \frac{1+z_o}{\delta_{ij}(1+z_c)} \tag{14}$$

4. Standard Display Mode

While some might argue that the above equations speak for themselves, we believe it is very instructive to consider the qualitative implications of their use – that is, a description of what exactly we see when we "move" the camera in the Java application. For the sake of completeness, we will also detail a number of cosmological effects that have been omitted from the application due to technical limitations. An example of the standard display mode is shown in Figure 3.

When we move the camera to a certain position in the HUDF data cube, we are in general viewing the Universe as it would appear from that point and at that redshift. We must qualify this statement by noting that the simulation accounts *only* for cosmological effects of changing the camera position – no dynamical, lensing, evolutionary, or other effects are simulated. In this sense, AHaH thus truly, though hypothetically, allows the user to travel through the Universe at "hyper-speed."

The somewhat counterintuitive relationship between an object's angular size and its redshift is readily apparent in the standard display mode. If a user slowly increases the redshift of the camera, high redshift objects will begin to decrease in angular size and move toward the center of the display, eventually reaching a minimum angular size and then increasing. Also visible are the effects of galaxy evolution and merging over time. For example, when viewing the Universe from redshift $z = 0.5$ as in Figure 3, there are many large spiral and elliptical galaxies visible. However, when viewing the Universe from redshift $z = 1.5$ as in Figure 4, the screen is dominated by small and compact blue galaxies.

It should be noted that the application does not make calculations for cosmological surface brightness dimming or changes in color due to redshift or spectral evolution. While certainly feasible to simulate, performing such image manipulation techniques on large numbers of galaxies in real-time is currently too difficult for consumer computers. Moreover, we must also recall that the HUDF data are limited in both magnitude and effective horizon by what could be observed from low Earth orbit. When we view the data from redshifts other than zero, we would expect to see more galaxies overall – including fainter galaxies – than are represented in the current HUDF data. We could choose to simulate these objects as extensions of our data set if we desired, but we felt this would not be particularly instructive, and could lead to potential confusion. Moreover, such simulations have a high degree of uncertainty and, by significantly increasing the size of the data set, would add prohibitively to the computation times. Likewise, we have chosen not to simulate galaxies outside of the original field, which would of course enter the camera's field of view as the user pans around.

5. Static Geometry Mode

When a user presses the "G" key in the Java tool, they are told that they are viewing the simulation with "static geometry" turned on. What this means specifically is that angular sizes as derived above are no longer affected by the scale factor or curvature of the Universe – after we develop our original coordinates, as in Equation (10), all calculations for angles are simply done with $\theta = \theta_E$. This has the visual effect of all galaxies appearing smaller and closer to the center of the viewport, since all initial angles have been contracted by a factor of $(1 + z)$ (when Ω_K is zero). In this static case, galaxies will also simply increase in angular size as we approach them, as opposed to the angular sizes of high-redshift objects in the real WMAP Universe, which decrease, reach a minimum, and then increase as the camera's redshift increases.

This static mode of viewing the simulation has no physical analogue – it is simply meant to convey to the user that there are non-Euclidean aspects of the Universe's geometry, and that the angular sizes we observe in the present have been made larger due to the Universe's expansion. One should note that this display mode only considers expansion as it relates to angular size – the comoving radial distance is still calculated using the redshift and curvature factors that would not be present in a strictly Euclidean Universe. That is, in the static display mode, we assume that the Hubble Law distance, $D = v/H_0 = (c/H_0)z$, is simply a Euclidean distance unrelated to expansion. This is primarily because our method of calculating the comoving radial distance relies upon redshift, which is a phenomenon specific only to an expanding Universe, and is therefore the only way we could calculate distances for all galaxies.

6. Conclusion

We believe that this software provides students and instructors with an unprecedented ability to interactively visualize many of the effects of an expanding Universe, among its other capabilities. The application should help clarify these concepts, and allow students to develop a deeper intuitive understanding of the material. Certain cosmological effects – such as bandpass shifting, k-correction, surface brightness dimming, gravitational lensing, and the effects of the magnitude limit and object sizes on the sample completeness limit – have been omitted due to computational limitations, but we believe these to be inessential for the understanding of the included effects. For a discussion of most of these effects, see e.g. Windhorst, Hathi, Cohen, & Jansen (2007).

For the convenience of those who wish to see or modify the particular implementation of the above formulae within the Java software, we have provided source code with the standard distribution of the tool. It is included in the src/ directory of ahah.jar and may be extracted using the java `jar` utility or any zlib-compatible de-compressor such as `unzip`. The tool may be downloaded from the AHaH website[10].

We thank Ned Wright for helpful discussion early in the project. We acknowledge the support from the Arizona State University NASA Space Grant (to MRM). This work was supported by grant XXXXX from the Space Telescope Science Institute, which is operated by AURA under NASA contract NAS 5-26555.

A. AHaH User Manual

- System Requirements
- Installing and Running the Application
- Main Screen and User Interface
- Moving the Camera
- Hotkeys and Special Functions
- Changing Options and Settings
- Configuration File and Advanced Options
- License, Source Code, and Modifications

System Requirements

The following are the minimum system requirements to run the Appreciating Hubble at Hyper-speed application:

- Microsoft Windows, Mac OS, or *nix operating system with Sun Java runtime version 1.4.3 or later
- 1.0 GHz processor
- 256 MB RAM
- Mouse and Keyboard

We recommend the following for optimal system performance:

- 2.0 GHz dual-core processor
- 1.0 GB RAM
- Graphics accelerator card with 64 MB video RAM

Additionally, an internet connection is required if you wish to view

---

[10] http://www.asu.edu/clas/hst/www/ahah/

extended information about selected galaxies.

Installing and Running the Application

The Appreciating Hubble at Hyper-speed application is distributed as an archive containing a standard executable Java resource file. To install the application, simply download the archive file for your operating system and save it wherever you prefer. Extract the archive wherever you like and then double-click the ahah.jar file to run the program. If you prefer to run from the command line, use java -jar ahah.jar.

Also provided is a configuration file, ahah.conf, that provides several options that you may modify if you want to use custom settings every time you run the program. See the Configuration File section for more information.

Main Screen and User Interface

The main screen represents your primary interface with the application. Besides holding the actual visualization, the screen also provides various forms of information about the simulation and methods to interact with and gain more information about specific galaxies.

The numbered user interface elements are as follows:

1. Position Indicator, showing the camera's x and y position within the coordinate system, as well as the camera's redshift.
2. Active Geometry Indicator, showing whether the Universe's real geometry is active or a non-physical geometry in which angular sizes are unaffected by the Universe's expansion.
3. Info Box that shows more information about the selected galaxy, including its HUDF ID, redshift, and comoving radial distance.
4. Jump Dialog, where you may enter an Object ID or redshift on which to center the camera.
5. Speed Indicator. Used to remind you that the speeds represented in the application (when moving) are not physically attainable.

Moving the Camera

You may move the camera using either the mouse or keyboard, or with a combination of both. We recommend that most users navigate with the mouse, though advanced users may find the precision movement afforded by the keyboard useful in some situations.

To navigate using the mouse, simply click and hold the left mouse button anywhere within the Main Screen. Moving the mouse will then move the camera left, right, up, and down. To move forward and backward, either scroll the mouse wheel up and down, or click and hold the mouse wheel or middle mouse button, then move the mouse forward and backward. You may also move forward and backward by holding shift while clicking and holding the left mouse button.

To navigate using the keyboard, simply use the arrow keys to move left, right, up, and down. To move forward and backward, hold shift and use the up and down arrow keys. Holding control while using keyboard navigation will move the camera at 10x the normal speed.

In addition to direct navigation, you may double left-click on any galaxy to initiate an automatic move to it. This will move the camera at a moderate pace along the line (geodesic) between the camera and the galaxy. The automatic move can be cancelled at any time by performing most other actions, such as attempting to move the camera or selecting a galaxy. Note that toggling the spatial geometry mode will not cancel an automatic movement.

Hotkeys and Special Functions

The application contains a number of additional features besides moving the camera, most of which are accessed using hotkeys. The following table contains an overview of these additional features and their hotkeys:

Program Feature	Hotkey
Select a galaxy and open its Info Box	Left-click galaxy
Toggle spatial geometry mode	G
Open Jump Dialog to travel to a specific galaxy or redshift	J
Reset the simulation and move the camera to the origin	R or F5
Open the Help and Options Dialog	H or F1

Info Box

The Info Box contains various information about the selected galaxy:

* The Object ID is a number uniquely identifying each galaxy and can be used in conjunction with the Jump Dialog to direct another user to a specific galaxy.

* The Redshift represents the factor by which a galaxy's observed spectrum is shifted from what we would observe if it were nearby. It is this measurement that allows us to calculate distances to galaxies.

* The Comoving Radial Distance represents the spatial separation of the galaxy from the origin at the present epoch, assuming zero peculiar velocity with respect to our own galaxy. It is calculated using the currently active cosmology constants.

* The Stamp Size is the length of one side of the black box surrounding the galaxy's picture. Thus if a galaxy takes up roughly half the width of the box and has a stamp size of 30 kpc, its proper transverse diameter should be about 15 kpc.

* The More Information Link links to a web page containing additional data for the selected galaxy. This includes its position (RA and Dec) and spectral fitting data. Note that not all galaxies have this additional information available. Specifically, it is available only for those galaxies that were included in the GRAPES survey, which are most of the bright, large galaxies.

* The Jump Button moves the camera to a view centered on the selected galaxy, exactly as if the galaxy's ID number had been entered into the Jump Dialog.

Spatial Geometry Mode

Toggling the spatial geometry mode changes between the Universe's actual geometry and a geometry where angular sizes are unaffected by the expansion of the Universe. This second geometry does not have a physical analogue - it does not represent what the Universe would look like if it were not expanding, or any other such set of circumstances. Instead, it is provided simply to convey visually that observed angular sizes have been affected by the scale factor.

Jump Dialog

The Jump Dialog is a way to quickly move to a specific location. If you enter an Object ID number then the simulation will move the camera to a position centered on that object. This is useful for labs that require students to examine a number of specific galaxies. If no Object ID is entered then the simulation will move the camera to the specified redshift, leaving the x and y coordinates fixed.

Resetting

Using the reset function will reset the simulation, exactly as if the application had been newly opened. This includes reloading all galaxies - both images and position data - as well as recalculating all distances and moving the camera to the origin. It also resets the graphics rendering thread, so may be used in the event that images fail to update.

Help and Options Dialog

The Help and Options Dialog gives you basic information about running the application, such as movement controls, as well as allowing you to modify some of the simulation's configuration parameters at runtime. It is discussed in detail in the next section. Additionally, the license for the software is also provided in this dialog.

Changing Options and Settings

There are two primary ways of changing options and settings for the application. These are the Configuration File, which will be covered in the next section, and the options pane of the Help and Options Dialog. The options pane contains two primary sections, Cosmology Parameters and Program Performance.

Cosmology Parameters

The Cosmology Parameters section allows you to modify certain constants of the displayed cosmology. There are currently four parameters that may be modified at runtime:

* H0 is the Hubble Constant, which goes into Hubble's Law relating recessional velocity and distance. It is also used when obtaining proper distances from coordinate distances within the application. Its units are km/s/Mpc.

* M is the matter density parameter of the Universe. It represents what fraction of the total energy density of the Universe is attributable to matter, both baryonic and dark.
* is the so-called vacuum energy density parameter of the Universe. It represents what fraction of the total energy density of the Universe is attributable to dark energy, ie. expansion energy associated with the Cosmological Constant, .
* R is the radiation density parameter of the Universe. It represents what fraction of the total energy density of the Universe is attributable to radiation.

The curvature parameter, K is derived from these supplied parameters internally as K = 1 - (M + + R). Most changes to these parameters will result in rather subtle visual effects - they will manifest primarily as changes to the comoving radial distance in a galaxy's Info Box. Curvatures that are very large in magnitude (|K| >> 1) may result in more marked visual differences.

Program Performance

The Program Performance section contains one value that may be changed at runtime - the Cull Size. It sets the smallest size in pixels for which a galaxy's image will be displayed - if a galaxy's size is smaller than the Cull Size, the galaxy will be displayed as a single pixel of its averaged color. Modification of this value is provided as a way to adjust the speed at which the application runs. The most CPU-intensive task is resizing and displaying images, so by increasing the Cull Size you may decrease the number of displayed images and thus increase the speed of the simulation. Note that decreasing the Cull Size will provide better visuals but could potentially slow the application down significantly.

Configuration File and Advanced Options

In addition to runtime options in the Help and Options Dialog, you may create or download an optional configuration file to modify the application's default behavior. This file is simply a specially formatted text file called ahah.conf that must be placed in the same directory as ahah.jar. The options are placed in the file as Key : Value pairs separated by a colon. For example the line OmegaM : 0.237 sets M to 0.237, the currently accepted COBE/WMAP value. Comments within the file are preceded by

and section headings (optional) are placed in brackets, such as [Cosmology Constants]. Keys other than those below will simply be ignored, and giving a key an improper value (such as a floating point number where an integer is required) will cause the application to print an error and then use the default value. The available options are as follows:

Application Operation Options

Options in this section change the default behaviors of how the application runs, including performance tweaks.

* FrameSleepTime is an integer that specifies the time in milliseconds to wait between redraws of the main simulation window. A shorter time will update the viewport more often, but be more taxing on the CPU. Modifying the default value may affect performance drastically.
* MacroStepDelay is an integer that specifies the time in milliseconds to wait between steps of an automatic move (when double-clicking on a galaxy). A shorter time will result in smoother, faster movement.
* CullSize is an integer that specifies the default Cull Size, as discussed in the Options and Settings section above. This way if you have a particularly fast computer, you can force the application to use a smaller Cull Size every time you run it. Thus you would not have to open the Help and Options Dialog to change it every time. Note that modifying the value at runtime using the Help and Options Dialog will always override any value in the configuration file.
* DefaultBrowser is a string representing your preferred browser command on *nix systems. If you do not change the setting yourself, the application will try, in order: firefox, mozilla, opera, konqueror, epiphany, netscape. The value set in the configuration file is simply any shell command, so may be something in $PATH like firefox, or the full path such as /usr/local/share/firefox/firefox. On Windows and Mac OS, the operating system's default browser is always used, so modifying this setting will have no effect.

Cosmology Constants

Options in this section set the defaults for various cosmology constants. For more detailed explanations, see the descriptions for the parameters in the Options and Settings section above. Like the Cull Size, changing the values at runtime using the Help and Options Dialog will override settings in the configuration file.

* HNought is a floating point number that specifies the default value of the Hubble Constant, H0.
* OmegaM is a floating point number that specifies the default value of the matter density parameter, M.
* OmegaV is a floating point number that specifies the default value of the vacuum energy density parameter, .
* OmegaR is a floating point number that specifies the default value of the radiation density parameter, R.

Dataset Information

Options in this section give the application information about the original image that the individual stamps come from, as well as specifying where to find the object database. Options from this section should generally not be changed by users - they are provided for future use of the application with different data.

* DBFile is a string representing the path of the database file within the jar file. This should only be modified if the application is repackaged with different data. The file should be specified with the root of the path being the root of the jar file, not of the computer's filesystem.
* ImageSizePx is an integer specifying the height and width of the original image in pixels. If for some reason the image is not square, use whichever dimension is larger.
* ImageArcsecPerPx is a floating point number that specifies the arcsec/px scale of the original image. This is used internally for changing pixel sizes to angular sizes and vice versa.
* MaxRedshift in an integer that specifies the maximum redshift for the distance tables. This need only be sufficiently large such that no object in the database has a higher redshift (as the program may then crash since it would not be able to correctly interpolate the object's distance).

Application Configuration Constants

Options in this section change aspects of the application that users generally do not need to modify, such as the locations of external resources like this documentation. These options are provided primarily as a convenience to website administrators in the event that urls change, etc.

- StampURLBase is a string that specifies the directory containing the More Information html files for galaxies. The provided string must contain the trailing /. The application expects the html files within this directory to be named n.html, where n is the HUDF ID number of the object.
- HelpURL is a string that indicates the url of the application's online documentation, i.e. this page.
- TextColor is a hex code that defines the color of the text on the Main Screen. The value is standard RGB hex encoding (like in HTML and CSS), i.e. RRGGBB. For instance a deep purple would be encoded as TextColor : 6633aa.

License, Source Code, and Modifications

The Appreciating Hubble at Hyper-speed application is provided under a BSD-like license. In simple terms, this means that virtually any modification or redistribution of the application is permitted, with the following caveats:

- Any redistribution must retain the original copyright notice and license file, either with the source code or with the documentation in the case of binary distributions.
- The names of the copyright holders, contributors, and associated institutions may not be used to endorse or promote any derivative works without prior permission.

Note that there is no requirement that source code be provided with any derivative work or redistribution. We encourage any derivative works to provide source code as well so that others may learn from your modifications, but we leave such decisions to your discretion.

Application Source Code

The application's source code is provided with the standard distribution. Java archives are simply standard ZIP (DEFLATE) archives with a special internal directory structure. Thus, you need only extract the contents of the archive using your favorite ZIP extractor (Using jar you would run jar -xf ahah.jar). The application source code is contained in the src/ directory.

Warning: The ahah.jar file exhibits "tarbomb" behavior, extracting to the current directory instead of its own subdirectory. This is mostly a limitation of the jar format, since the META-INF/ directory must always be in the archive root. We strongly suggest you move the archive to its own directory before extracting it.

To compile the source you simply need to run javac *.java. The compiled binaries will automatically be placed within the directory edu/asu/Ahah/. To run the modified application you then need to create a new jar including the directories edu/, images/, and data/, as well as the files defaults.conf and license.txt, and any source code if you desire. You also need to specify a Manifest file, which must contain at least the line Main-Class: edu.asu.Ahah.Ahah - this tells the Java interpreter where to find the main class when a user runs the jar. An example jar creation command is:

```
jar -cvfm ahah.jar MANIFEST.MF edu images data src defaults.conf license.txt
```

REFERENCES

Beckwith, S. V. W., et al. 2006, AJ, 132, 1729

Bolzonella, M., Miralles, J.-M., & Pelló, R. 2000, A&A, 363, 476

Bouwens, R. J., Illingworth, G. D., Blakeslee, J. P., & Franx, M. 2006, ApJ, 653, 53B

Longair, M. S. 1998, Galaxy Formation (Berlin: Springer-Verlag)

Lupton, R., Blanton, M. R., Fekete, G., Hogg, D. W., O'Mullane, W., Szalay, A., & Wherry, N. 2004, PASP, 116, 133

Pirzkal, N., et al. 2004, ApJS, 154, 501

Ryan, R. E., Jr., et al. 2007, ApJ, in press

Ryden, B. 2003, Introduction to Cosmology (San Francisco, CA: Addison-Wesley)

Thompson, R. I., et al. 2005, AJ, 130, 1

Windhorst, R. A., Hathi, N. P., Cohen, S. H., & Jansen, R. A. 2007, in the 36th COSPAR Scientific Assembly

Wright, E. L. 2006, preprint (astro-ph/0609593v2)

Yan, H. & Windhorst, R. A. 2004, ApJ, 612, L93a

This preprint was prepared with the AAS LaTeX macros v5.2.

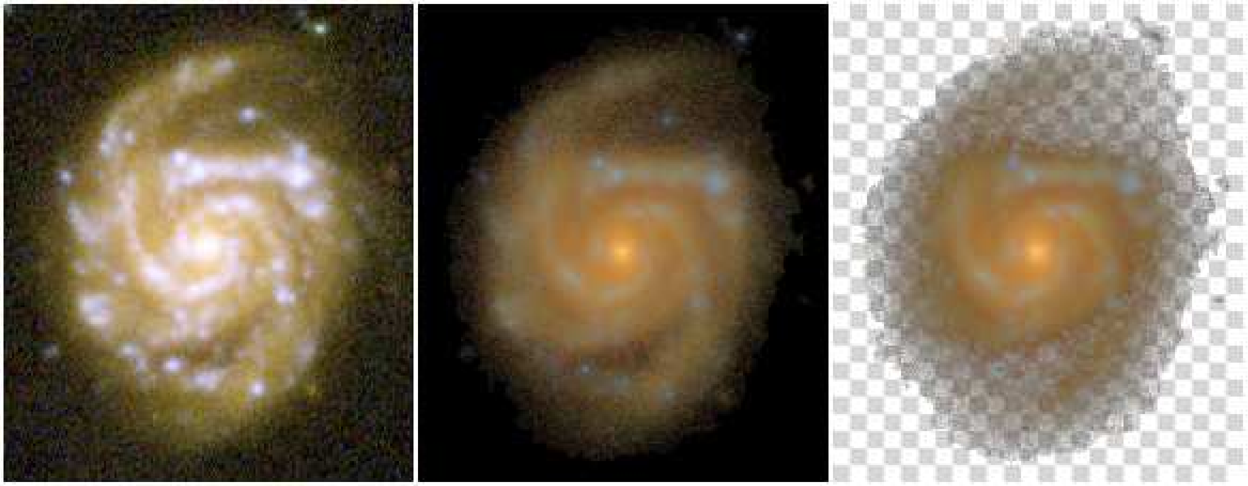


Fig. 1.— A comparison of three images of HUDF galaxy 7556. The left image is that from the original STScI release, clearly showing the bright, burnt-out knots characteristic of the standard logarithmic image stretch. The center image is our prepared image using the arcsinh stretch described by Lupton et al. (2004), as it appears in the AHaH application. The right image is our prepared image against an artificially imposed chessboard pattern, showing the included transparency. Note that pixels outside the source are all completely transparent, since they have been removed entirely using the `SExtractor` segmentation map.

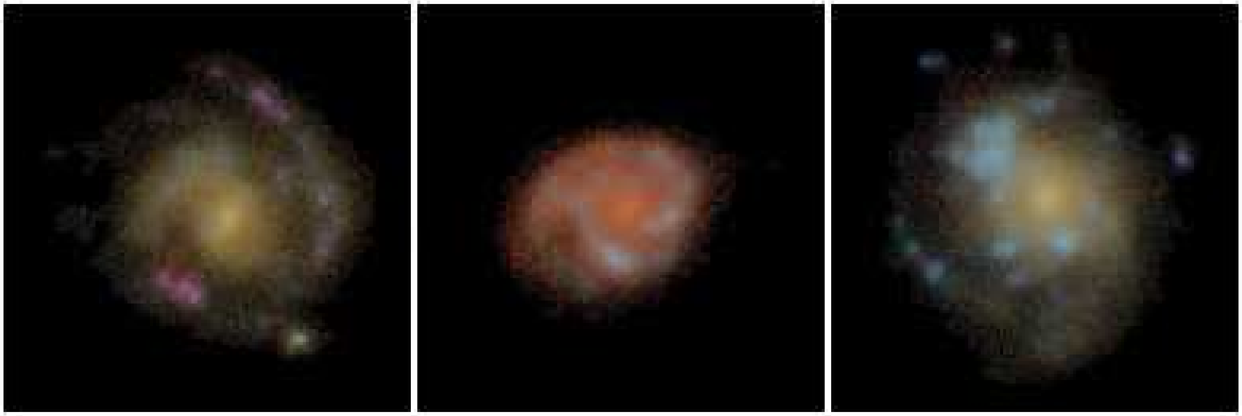
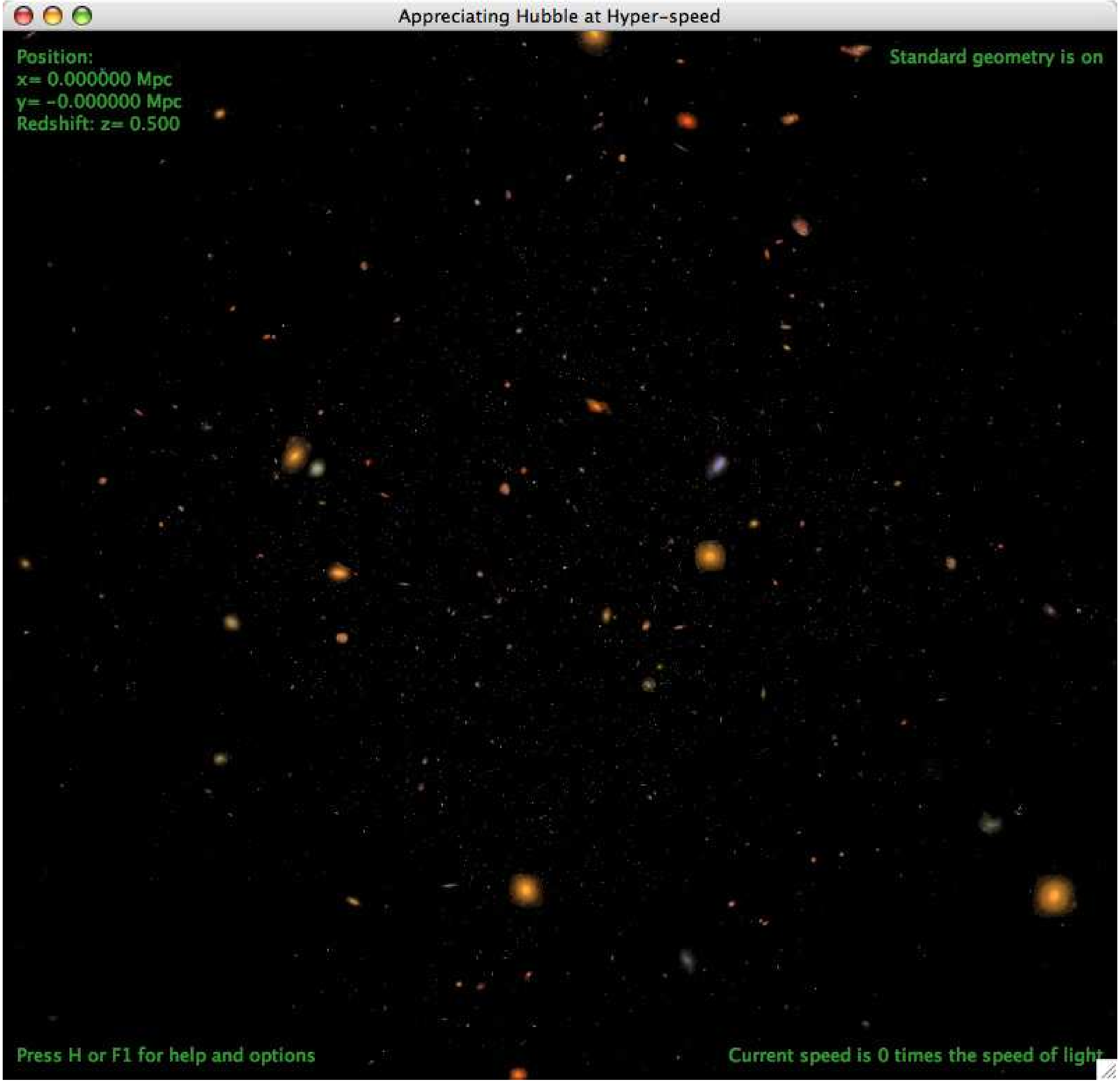


Fig. 2.— Our prepared images of three galaxies from the HUDF, using the arcsinh stretch described by Lupton et al. (2004). Shown are galaxy 3180 (left), galaxy 5805 (center), and galaxy 6974 (right).



Position:
x= 0.000000 Mpc
y= -0.000000 Mpc
Redshift: z= 0.500

Standard geometry is on

Press H or F1 for help and options

Current speed is 0 times the speed of light

Fig. 3.— The HUDF data as viewed from redshift $z = 0.5$ in the AHaH application, using standard geometry mode, which properly calculates angular sizes. Note how the image is dominated by luminous red early-type galaxies.

Appreciating Hubble at Hyper-speed

Position:
x= 0.000000 Mpc
y= -0.000000 Mpc
Redshift: z= 1.500

Standard geometry is on

Press H or F1 for help and options

Current speed is 0 times the speed of light

Fig. 4.— The HUDF data as viewed from redshift $z = 1.5$ in the AHaH application, using standard geometry mode. Note how this image is dominated by blue irregular and merging star-forming galaxies.