Ppmforge User Manual

ppmforge

Updated: 25 October 1991
Table Of Contents

NAME

ppmforge - fractal forgeries of clouds, planets, and starry skies

SYNOPSIS

ppmforge [**-clouds**] [**-night**] [**-dimension** *dimen*] [**-hour** *hour*] [**-inclination**|**-tilt** *angle*] [**-mesh** *size*] [**-power** *factor*] [**-glaciers** *level*] [**-ice** *level*] [**-saturation** *sat*] [**-seed** *seed*] [**-stars** *fraction*] [{**-xsize**|**-width**} *width*] [{**-ysize**|**-height**} *height*]

You can abbreviate any options to its shortest unique prefix.

DESCRIPTION

This program is part of Netpbm. **ppmforge** generates three kinds of ''random fractal forgeries,'' the term coined by Richard F. Voss of the IBM Thomas J. Watson Research Center for seemingly realistic pictures of natural objects generated by simple algorithms embodying randomness and fractal self-similarity. The techniques used by **ppmforge** are essentially those given by Voss[1], particularly the technique of spectral synthesis explained in more detail by Dietmar Saupe[2].

The program generates two varieties of pictures: planets and clouds, which are just different renderings of data generated in an identical manner, illustrating the unity of the fractal structure of these very different objects. A third type of picture, a starry sky, is synthesised directly from pseudorandom numbers.

The generation of planets or clouds begins with the preparation of an array of random data in the frequency domain. The size of this array, the ''mesh size,'' can be set with the **-mesh** option; the larger the mesh the more realistic the pictures but the calculation time and memory requirement increases as the square of the mesh size. The fractal dimension, which you can specify with the **-dimension** option, determines the roughness of the terrain on the planet or the scale of detail in the clouds. As the fractal dimension is increased, more high frequency components are added into the random mesh.

Once the mesh is generated, an inverse two dimensional Fourier transform is performed upon it. This converts the original random frequency domain data into spatial amplitudes. We scale the real components that result from the Fourier transform into numbers from 0 to 1 associated with each point on the mesh. You can further modify this number by applying a ''power law scale'' to it with the **-power** option. Unity scale leaves the numbers unmodified; a power scale of 0.5 takes the square root of the numbers in the mesh, while a power scale of 3 replaces the numbers in the mesh with their cubes. Power law scaling is best envisioned by thinking of the data as representing the elevation of terrain; powers less than 1 yield landscapes with vertical scarps that look like glacially-carved valleys; powers greater than one make fairy-castle spires (which require large mesh sizes and high resolution for best results).

After these calculations, we have a array of the specified size containing numbers that range from 0 to 1. The pixmaps are generated as follows:

CloudsA color map is created that ranges from pure blue to white by increasing admixture (desaturation) of blue with white. Numbers less than 0.5 are colored blue, numbers between 0.5 and 1.0 are colored with corresponding levels of white, with 1.0 being pure white.

Planet The mesh is projected onto a sphere. Values less than 0.5 are treated as water and values between 0.5 and 1.0 as land. The water areas are colored based upon the water depth, and land based on its elevation. The random depth data are used to create clouds over the oceans. An atmosphere approximately like the Earth's is simulated; its light absorption is calculated to create a blue cast around the limb of the planet. A function that rises from 0 to 1 based on latitude is modulated by the local elevation to generate polar ice caps--high altitude terrain carries glaciers farther from the pole. Based on the position of the star with respect to the observer, the apparent color of each pixel of the planet is calculated by ray-tracing from the star to the planet to the observer and applying a lighting model that sums ambient light and diffuse reflection (for most planets ambient light is zero, as their primary star is the only source of illumination). Additional random data are used to generate stars around the planet.

Night A sequence of pseudorandom numbers is used to generate stars with a user specified density.

Cloud pictures always contain 256 or fewer colors and may be displayed on most color mapped devices without further processing. Planet pictures often contain tens of thousands of colors which must be compressed with **pnmquant** or **ppmdither** before encoding in a color mapped format. If the display resolution is high enough, **ppmdither** generally produces better looking planets. **pnmquant** tends to create discrete color bands, particularly in the oceans, which are unrealistic and distracting. The number of colors in starry sky pictures generated with the **-night** option depends on the value specified for **-saturation**. Small values limit the color temperature distribution of the stars and reduce the number of colors in the image. If the **-saturation** is set to 0, none of the stars will be colored and the resulting image will never contain more than 256 colors. Night sky pictures with many different star colors often look best when color compressed by **pamdepth** rather than **pnmquant** or **ppmdither**. Try *newmaxval* settings of 63, 31, or 15 with **pamdepth** to reduce the number of colors in the picture to 256 or fewer.

OPTIONS

-clouds Generate clouds. A pixmap of fractal clouds is generated. Selecting clouds sets the default for fractal dimension to 2.15 and power scale factor to 0.75.

-dimension *dimen* Sets the fractal dimension to the specified *dimen*, which may be any floating point value between 0 and 3. Higher fractal dimensions create more ‘‘chaotic’’ images, which require higher resolution output and a larger FFT mesh size to look good. If no dimension is specified, 2.4 is used when generating planets and 2.15 for clouds.

-glaciers *level* The floating point *level* setting controls the extent to which terrain elevation causes ice to appear at lower latitudes. The default value of 0.75 makes the polar caps extend toward the equator across high terrain and forms glaciers in the highest mountains, as on Earth. Higher values make ice sheets that cover more and more of the land surface, simulating planets in the midst of an ice age. Lower values tend to be boring, resulting in unrealistic geometrically-precise ice cap boundaries.

-hour *hour* When generating a planet, **ppmforge** uses *hour* as the "hour angle at the central meridian." If you specify **-hour 12**, for example, the planet will be fully illuminated, corresponding to high noon at the longitude at the center of the screen. You can specify any floating point value between 0 and 24 for *hour*, but values which place most of the planet in darkness (0 to 4 and 20 to 24) result in crescents which, while pretty, don't give you many illuminated pixels for the amount of computing that's required. If no **-hour** option is specified, a random hour angle is chosen, biased so that only 25% of the images generated will be crescents.

-ice *level* Sets the extent of the polar ice caps to the given floating point *level*. The default level of 0.4 produces ice caps similar to those of the Earth. Smaller values reduce the amount of ice, while larger **-ice** settings create more prominent ice caps. Sufficiently large values, such as 100 or more, in conjunction with small settings for **-glaciers** (try 0.1) create "ice balls" like Europa.

-inclination|**-tilt** *angle* The inclination angle of the planet with regard to its primary star is set to *angle*, which can be any floating point value from -90 to 90. The inclination angle can be thought of as specifying, in degrees, the ‘‘season’’ the planet is presently experiencing or, more precisely, the latitude at which the star transits the zenith at local noon. If 0, the planet is at equinox; the star is directly overhead at the equator. Positive values represent summer in the northern hemisphere, negative values summer in the southern hemisphere. The Earth's inclination angle, for example, is about 23.5 at the June solstice, 0 at the equinoxes in March and September, and -23.5 at the December solstice. If no inclination angle is specified, a random value between -21.6 and 21.6 degrees is chosen.

-mesh *size*A mesh of *size* by *size* will be used for the fast Fourier transform (FFT). Note that memory requirements and computation speed increase as the square of *size*; if you double the mesh size, the program will use four times the memory and run four times as long. The default mesh is 256x256, which produces reasonably good looking pictures while using half a megabyte for the 256x256 array of single precision complex numbers required by the FFT. On machines with limited memory capacity, you may have to reduce the mesh size to avoid running out of RAM. Increasing the mesh size produces better looking pictures; the difference becomes particularly noticeable when generating high resolution images with relatively high fractal dimensions (between 2.2 and 3).

-nightA starry sky is generated. The stars are created by the same algorithm used for the stars that surround planet pictures, but the output consists exclusively of stars.

-power *factor*Sets the "power factor" used to scale elevations synthesised from the FFT to *factor*, which can be any floating point number greater than zero. If no factor is specified a default of 1.2 is used if a planet is being generated, or 0.75 if clouds are selected by the **-clouds** option. The result of the FFT image synthesis is an array of elevation values between 0 and 1. A non-unity power factor exponentiates each of these elevations to the specified power. For example, a power factor of 2 squares each value, while a power factor of 0.5 replaces each with its square root. (Note that exponentiating values between 0 and 1 yields values that remain within that range.) Power factors less than 1 emphasise large-scale elevation changes at the expense of small variations. Power factors greater than 1 increase the roughness of the terrain and, like high fractal dimensions, may require a larger FFT mesh size and/or higher screen resolution to look good.

-saturation *sat*Controls the degree of color saturation of the stars that surround planet pictures and fill starry skies created with the **-night** option. The default value of 125 creates stars which resemble the sky as seen by the human eye from Earth's surface. Stars are dim; only the brightest activate the cones in the human retina, causing color to be perceived. Higher values of *sat* approximate the appearance of stars from Earth orbit, where better dark adaptation, absence of skyglow, and the concentration of light from a given star onto a smaller area of the retina thanks to the lack of atmospheric turbulence enhances the perception of color. Values greater than 250 create ‘‘science fiction’’ skies that, while pretty, don't occur in this universe.

Thanks to the inverse square law combined with Nature's love of mediocrity, there are many, many dim stars for every bright one. This population relationship is accurately reflected in the skies created by **ppmforge**. Dim, low mass stars live much longer than bright massive stars, consequently there are many reddish stars for every blue giant. This relationship is preserved by **ppmforge**. You can reverse the proportion, simulating the sky as seen in a starburst galaxy, by specifying a negative *sat* value.

-seed *num*Sets the seed for the random number generator to the integer *num*. The seed used to create each picture is displayed on standard output (unless suppressed with the **-quiet** option). Pictures generated with the same seed will be identical. If no **-seed** is specified, a random seed derived from the date and time will be chosen. Specifying an explicit seed allows you to re-render a picture you particularly like at a higher resolution or with different viewing parameters.

-stars *fraction*Specifies the percentage of pixels, in tenths of a percent, which will appear as stars, either surrounding a planet or filling the entire frame if **-night** is specified. The default *fraction* is 100.

-xsize|**-width** *width*Sets the width of the generated image to *width* pixels. The default width is 256 pixels. Images must be at least as wide as they are high; if a width less than the height is specified, it will be increased to equal the height. If you must have a long skinny pixmap, make a square one with **ppmforge**, then use **pamcut** to extract a portion of the shape and size you require.

-ysize|**-height** *height*Sets the height of the generated image to *height* pixels. The default height is 256 pixels. If the height specified exceeds the width, the width will be increased to equal the height.

LIMITATIONS

The algorithms require the output pixmap to be at least as wide as it is high, and the width to be an even number of pixels. These constraints are enforced by increasing the size of the requested pixmap if necessary.

You may have to reduce the FFT mesh size on machines with 16 bit integers and segmented pointer architectures.

SEE ALSO

pamcut, **pamdepth**, **ppmdither**, **pnmquant**, **ppm**

[1] Voss, Richard F., ''Random Fractal Forgeries,'' in Earnshaw et. al., Fundamental Algorithms for Computer Graphics, Berlin: Springer-Verlag, 1985.

[2] Peitgen, H.-O., and Saupe, D. eds., The Science Of Fractal Images, New York: Springer Verlag, 1988.

AUTHOR

John Walker
Autodesk SA
Avenue des Champs-Montants 14b
CH-2074 MARIN
Suisse/Schweiz/Svizzera/Svizra/Switzerland
Usenet:kelvin@Autodesk.com
Fax:038/33 88 15
Voice:038/33 76 33

PLUGWARE!

If you like this kind of stuff, you may also enjoy ''James Gleick's Chaos--The Software'' for MS-DOS, available for $59.95 from your local software store or directly from Autodesk, Inc., Attn: Science Series, 2320 Marinship Way, Sausalito, CA 94965, USA. Telephone: (800) 688-2344 toll-free or, outside the U.S. (415) 332-2344 Ext 4886. Fax: (415) 289-4718. ''Chaos--The Software'' includes a more comprehensive fractal forgery generator which creates three-dimensional landscapes as well as clouds and planets, plus five more modules which explore other aspects of Chaos. The user guide of more than 200 pages includes an introduction by James Gleick and detailed explanations by Rudy Rucker of the mathematics and algorithms used by each program.

Table Of Contents

OPTIONS
LIMITATIONS
SEE ALSO
AUTHOR