Synthesis of a scale invariant texture on the sphere spherical_cpc

Emilie Koenig

16 avril 2010

Set of functions for the synthesis and the visualization of a scale invariant spherical texture.

This toolbox contains 8 folders :
- **Synthesis**/ : set of functions for the synthesis of a scale invariant texture directly on the sphere
- **Meshes**/ : functions used to create diverse spherical meshes
- **Visualization**/ : program to visualize a textured sphere using OpenGL
- **Filter**/ : a low-pass filter on the sphere preserving the scale invariance of the spherical texture
- **Spherical_Harmonics**/ : tool to visualize a zonal spherical harmonic of given level
- **Elevation_map**/ : set of functions to create a given elevation map. This map can be used in the visualization process to create an elevation independent of the texture
- **Torus**/ : creation of a torus to put on the sphere
- **Examples**/ : example of a scale invariant texture, its filtered version, the elevation pattern, a torus and a zonal spherical harmonic.
- **Documentation**/ : this folder, contains this description of spherical_cpc

It also contains 3 files :
- makefile : a makefile compiling all the programs used here
- INSTALL : a set of bash instructions to create bash paths for a Mac or a Linux system
- README : a brief description of this folder

This document only presents the contents of the principal folders (all but **Example**/).

Synthesis/

Synthesis of a scale invariant texture directly on a sphere. This synthesis is independent of the spherical mesh chosen because it is done in the continuous domain. This synthesis is an extension of the compound Poisson cascades developed in [1]. We will not describe here the mechanism of this synthesis. For more information see [2]. In tis document we will only focus on the use of the synthesis functions.

Use of the runnable program :
- minimal use :

  ./synthesis -f spherical_mesh
- complete use :

  ./synthesis -f spherical_mesh -Ray R -L L -r rmin -l law -p param1 param2 -m pattern -c C -v version -g seed

The diverse parameters and their default values are :
- spherical_mesh : spherical mesh to use for the synthesis. The meshes are located in the folder **Meshes**/.
- R : sphere radius, default R=1.
- L : larger scale of the texture, default L=1.
- rmin : smaller scale of the texture, default rmin=0.01.
- law : distribution law of the values of random i.i.d. variables, default law=2.
  - 0 : log-Poisson
  - 1 : log-normal
  - 2 : log-exponential
  - 3 : log-Bessel
- param1, param2 : parameters of the distribution law, default param1=0.4, param2=0.1
- pattern : pattern of the texture, default motif=1.
  - 0 : discus
  - 1 : shaper in square cosinus
  - 2 : shaper in exponential

  - 3 : other pattern (star)
- C : concentration of the i.i.d. random variables (multiply or divide the initial number), default C=1.
- version : kind of dilation chosen, default version=1.
  - 0 : dilation using a cone
  - 1 : angular dilation
  - 2 : stereographic dilation
- seed : initialization of the random variable generator, default=0.

This program gives the possibility to generate a set of independent spherical textures. For this, the number of textures to generate is required just after the launch of the program.

The program creates a file containing the coordinates of the spherical points and their values. It is named :

texture_ day_ hour

This file has the form :
- kind of file : 4 corresponds to a file containing the spherical points coordinates and their associated values.
- kind of spherical mesh :
  - 0 : subdivision mesh
  - 1 : equiangular mesh
  - 2 : HEALPix mesh
- points number :
  - for a subdivision mesh, we give the number of triangles used
  - for an equiangular mesh, we give the number of points by latitude
  - for a HEALPix mesh, we give the total number of points
- list of coordinates and their values

The program also provide a .log file with the same name as the texture file. It contains all the information about the spherical texture generated.

Meshes/

This folder contains 3 other folders to generate spherical meshes :
- **Equiangular/** : these meshes are non-uniform and have the same number of points at each latitudes
- **Subdivision/** : these meshes are created as the successive division of the faces of an elementary volume, the repartition of points is then uniform
- **HEALPix/** : mesh used mainly in the astrophysical domain, it is a uniform mesh where the center of faces are at constant latitudes

Each of these folders contains a folder named **Mesh/** containing the spherical meshes at diverse resolutions.

Visualization/

Visualization program using the libraries :
- avilib : used to create .avi movie files, provided
- glut and OpenGL : used for the visualization (creation of the window and the object, management of diverse options), to be install by the user (http ://www.opengl.org/resources/libraries/glut/ and http ://www.opengl.org/)
- libcfitsio : used to create .fits image files, provided (http ://heasarc.gsfc.nasa.gov/fitsio/)
- tr : used to create images of any size, even larger than the visualization window, provided (http ://www.mesa3d.org/brianp/TR.html)

Use of the runnable program :
- minimal use :

  ./visu -f texture_file
- complete use :

  ./visu -f texture_file -e elevation_file -t torus_file

The diverse parameters are :
- texture_file : texture file to be visualized
- elevation_file : elevation file if the user want an elevation independent of the texture
- torus_file : torus file

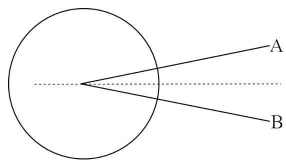
The program gives the possibility to control the texture in the visualization window. The keyboard commands are :

- x : display the cursor
- X : display the axes
- I : take a picture (of any size) of the object on the window
  - enter the size of the image
  - enter the number of images and the rotation angle between the images
  - enter the kind of image :
    - 0 : TGA image
    - 1 : FITSimage
  - name of the image taken :

    image_day_month_year_hour-image_number.tga / .fits
- V : take a .avi. Type 'V' to take the video, type 'V' another time to stop the video. The video file is named

  video_day_month_year_hour.avi
- space : center the camera
- Z : shift the camera to the top
- S : shift the camera to the bottom
- D : shift the camera to the right
- Q : shift the camera to the left
- r : rotation, needs to enter the angle
- z : rotation to the top
- s : rotation to the bottom
- d : rotation to the right
- q : rotation to the left
- E : elevation. The elevation values are given either by the texture values or by an independent elevation map. A percentage is needed, multiplied to the elevation values. Type 'E' to visualize the elevation, type 'E' another time to come back to a smooth visualization.
- A : process to take images automatically. This was developed to create a test bench.
  - enter the name of the image series, either « raycst » or « patte1 »
  - enter if the user want an elevation
  - enter the first number of the image serie
  - this function do an automatic rotation of the sphere to take quasi-independent textures. When an elevation is wanted, the elevation map stay at the same place on the sphere. A more precise explanation of this function is given later
- g : normalization of the texture values between 0 and 1 to visualize correctly the whole texture with OpenGL

Automatic image capture to create a test bench for the validation of Velociraptor [3] : This function create automatically a test bench for the validation of Velociraptor. From a spherical texture and an elevation map, this function creates a set of images of quasi-independent textures. To do this, the texture turns on the elevation map which stays fix. The test bench is created by taking two images (view A and B, taken on both sides of the texture center) separated by a given angle. The figure below presents the configuration used to take the views A and B :



This function generates, for a quasi-independent texture, a given series name and a given angle, images named

view_series_aangle_100pc_texturenumber.fits

where view is either A or B, series is the set name ('patte1' when an elevation map is used, 'raycst' otherwise), angle is the separation angle, texturenumber is the number of the texture used.

Filter/

This program realizes a pseudo-fractional integration on a scale invariant spherical texture. The pseudo-fractional integration is a low-pass filter in $1/k^H$ in the Fourier domain preserving the scale invariance of the texture. On the sphere, this filter is done in the spherical harmonics domain [4]. This needs to use an equiangular mesh.

This program needs the installation of the library fftw (http ://www.fftw.org/).

Use of the runnable program :

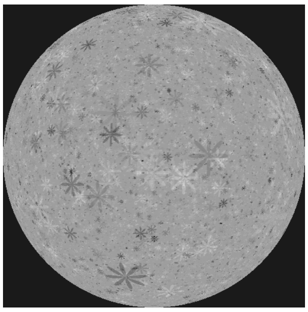
./filter -f texture_file -t filter_type

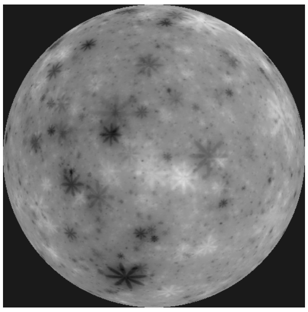
The diverse parameters are :
- texture_file : texture file to be filtered
- filter_type : type of filter to use
  - AR : back and forth in the spherical harmonics domain
  - PB : simple low-pass filter, need to enter the cut frequency
  - KH : pseudo-fractional integration in $1/k^H$, need to enter the H value

The program creates a texture file in the same folder than the initial texture file. It is named :

texture_day_hour_filter

The figure below gives an example of spherical texture and its filtered version :





Spherical_Harmonics/

This program creates a file to visualize a zonal spherical harmonics of a given level.

Use of the runnable program :

./Ylm -f mesh_file -l level

The diverse parameters are :
- mesh_file : spherical mesh file
- level : level of the zonal harmonic

The program creates a texture file named :

Ylm_day_hour

The figure below gives an example of a zonal spherical harmonic of level 15 :

Elevation_map/

This program creates a standard elevation map used for the validation of Velociraptor.

Use of the runnable program :

./map -f mesh_file -Ray R -r rmin -L L

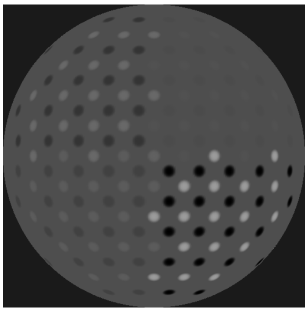
The diverse parameters and their default values are :

- mesh_file : equiangular mesh file used
- R : sphere radius, default R=1
- rmin : smaller scale of the map, default rmin=0.01
- L : larger scale of the map, default L=1 ;

The program creates a texture file named :

map_ day_ hour

The figure below gives an example of the elevation map produced :



Torus/

This program creates a torus.

Use of the runnable program :
- minimal use :

./tore -fout output

- complete use :

./tore -R R -r r -X nb1 -Y nb2 -Xc xc -Yc yc -Zc zc -a alpha -b beta -fout output

The diverse parameters and their default values are :
- output : name of the output file
- R : extern radius of the torus, default R=0.15.
- r : section radius o the torus, default r=0.015.
- nb1 : number of points on the torus section, default nb1=50.
- nb2 : number of points on the length of the torus, default nb2=200.
- xc, yc, zc : coordinates of the torus center, default (0,0,0).
- alpha : orientation angle on the Y axis, default alpha=0.
- beta : orientation angle on the X axis, default beta=0.

The output file can be visualize using the option -t of the visualization program.

Références

[1] P. Chainais, "Infinitely divisible cascades to model the statistics of natural images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 29, no. 12, pp. 2105–2119, 2007.

[2] E. Koenig, P. Chainais, "Multifractal analysis on the sphere," *Proceedings of ICISP'08*, pp 613–621, Cherbourg, 2008.

[3] S. Gissot, J.-F. Hochedez, P. Chainais, J.-P. Antoine "3D Reconstruction from SECCHI-EUVI Images Using an Optical-Flow Algorithm" *Solar Physics*, vol. 252, no. 2, pp 397–408, 2008.

[4] J. Driscoll, J. Healy "Computing Fourier transforms and convolutions on the 2-sphere" *Advances in Applied Mathematics*, vol. 15, pp 202–250, 1994.