CSCI 552(Spring 2021)

Homework #4

Name : Khushboo Mantri

Handout: Thursday, April 15, 2021

Due: 11:59 pm, Thursday, April 29, 2021

Total points: 50

All assignments will be submitted through Canvas. Documents will need to be in either Word or PDF format. Images need to be in jpeg format.

1. Consider a volume of dimensions 512×512×256. The attribute in each voxel is a 1-bytedensity value. Describe how you would represent this dataset using each of the following data formats, i.e. what information do you need to store in each data format? Approximately how much storage (in bytes) do you need in each case?

(a) structured points (volume), with cubical voxels.

(b) rectlinear grid

(c) structured grid

Home Work - 4

1) volume of dimensions 512×512×256.
The attribute in each voxel is a 1 byte density value. Following are the data formats on which we will represent data set.

a) structured points (volume), with cubical voxel - The file format support 1D, 2D, 3D structured points. The dimension nx, ny, nz must be greater than or equal to 1. The data spacing sx, sy, sz must be greater than 0.

Data structured points.
Dimension nx ny nz = n512 n512 n256
origin xyz
Spacing sx sy sz
∴ Byte required to store data = ~~6710~~ byte
= 67.108 mb aprox approx

b) Rectilinear grid - These are similar to regular grids in that the data is arranged along orthogonal axes, but the data need not to be evenly spaced ~~on~~ along the axis. The geometry is defined by three list of montonically increasing co-ordinate values, one list for each of the xyz co-ordinate axes. The topology of Rectilinear grid is defined by specifying the grid dimension, which is greater than or equal to 1

512 x 512 x 256 are array length of npksx, nptsy and nptsz. This is significant saving in memory. As total memory used is 512 + 512 + 256 = 1280 byte
= 0.00128 mb

c. Structured grid - These data point in these grids are regularly spaced but are not ~~allowed~~ aligned to ortogonal axes. The file format supports 1D, 2D, 3D structured grid. The dimensions nx, ny, nz must be greater than or equal to 1. The point coordinates are defined by the data in the points section.

Dimension nx ny nz
Points = P(n-1)x P(n-1)y P(n-1)z
origin = 0 (origin + index x spacing) = 2359296 byte
spacing = vector of length of 3
≈ 2.359 mb approx

Q2. Describe the linear interpolation procedure for tetrahedral cells using 3D barycentric coordinates, assuming scalar attributes are given at the four vertices of each tetrahedron. You need to compute the signed barycentric coordinates such that points outside the tetrahedron will have at least one negative coordinate.

Q2 Linear interpolation is a method of curve fitting using linear polynomials to construct new data points within the range of a discrete set of known data points

Barycentric co-ordinates may be easily extended to 3D. The 3D is a tetrahedron, ~~or~~ a polyhedron having 4 triangular faces and 4 vertices. The four barycentric co-ordinates are defined so that the first vertex r_1 map to barycentric co-ordinates $\lambda = (1,0,0,0)$, $r_2 \rightarrow (0,1,0,0)$ etc.

procedure for triangle to find barycentric co-ordinates of a point r with respect to a tetrahedron:

$$\begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{pmatrix} = T^{-1}(r - r_4)$$

$$T = \begin{pmatrix} x_1 - x_4 & x_2 - x_4 & x_3 - x_4 \\ y_1 - y_4 & y_2 - y_4 & y_3 - y_4 \\ z_1 - z_4 & z_2 - z_4 & z_3 - z_4 \end{pmatrix}$$

$\lambda_4 = 1 - \lambda_1 - \lambda_2$ with corresponding cartesian co-ordinates:

$$x = \lambda_1 x_1 + \lambda_2 x_2 + \lambda_3 x_3 + (1 - \lambda_1 - \lambda_2 - \lambda_3) x_4$$
$$y = \lambda_1 y_1 + \lambda_2 y_2 + \lambda_3 y_3 + (1 - \lambda_1 - \lambda_2 - \lambda_3) y_4$$
$$z = \lambda_1 z_1 + \lambda_2 z_2 + \lambda_3 z_3 + (1 - \lambda_1 - \lambda_2 - \lambda_3) z_4$$

Given a non-degenerated tetrahedron whose 4 vertices are non coplanar points $P_0, P_1, P_2, P_3 \in R^3$ and a point $P \in R^3$,

In barycentric co-ordinates
$$P = u_0 P_0 + u_1 P_1 + u_2 P_2 + u_3 P_3$$

Where $u_0, u_1, u_2, u_3 \in R$ and $u_0 + u_1 + u_2 + u_3 = 1$ knowing that all these coefficients sum up to 1.

Replace u_0 by $1 - u_1 - u_2 - u_3$

Transform the above equation to the following form.

$$u_1(P_1 - P_0) + u_2(P_2 - P_0) + u_3(P_3 - P_0) = (P - P_0)$$

$$= \cancel{P - P_0}$$

When all the coefficient computed

$\boxed{u_0, u_1, u_2, u_3 \text{ are positive} - P \text{ is inside the tetrahedron}}$

any of u_0, u_1, u_2, u_3 is $-ve$ - P is outside tetrahedron

one of u_0, u_1, u_2, u_3 is 0 and other $+ve$ - P is on a face of tetrahedron

two of u_0, u_1, u_2, u_3 are 0 and other $+ve$ - P is on the edge of tetrahedron

three of u_0, u_1, u_2, u_3 are 0 and other $+ve$ = P is a vertex of the tetrahedron

Q3. Consider a pulse function f(x) = 1 for−1≤x≤1 and f(x) = 0 elsewhere. Show that the Fourier transform off(x) is a multiple of the sinc function: sin(πx)/(πx) (i.e. filtering by a pulse function in the frequency domain is equivalent to a convolution using a sinc function in the spatial domain).

Q3 $f(x) = 1 \quad , \quad -1 \leq x \leq 1$

$= 0 \quad , \quad -\infty < x < 1 \qquad 1 < x < \infty$

Prove : Fourier transform of $f(x)$ is a multiple of the sinc function $\sin \frac{\pi x}{\pi x}$

∴ Fourier transform of $f(x)$

$$F\{f(x)\} = F(s) = \int_{-\infty}^{\infty} f(x) e^{isx} d(x)$$

$$= \int_{-\infty}^{-1} f(x) e^{isx} dx + \int_{-1}^{1} f(x) e^{isx} dx + \int_{1}^{\infty} f(x) e^{isx} dx$$

$$= \int_{-1}^{1} 1 \cdot e^{isx} dx = \left| \frac{e^{isx}}{is} \right|_{-1}^{1}$$

$$\sin x = \frac{e^{ix} - e^{-ix}}{2i}$$

$$\therefore F\{f(x)\} = F(s) = \frac{2 \sin s}{s}$$

Consider $s = \pi x$

$$\therefore F\{f(x)\} = \cancel{2\pi} \; \frac{2 \sin \pi x}{\pi x}$$

Scanned with CamScanner

Q4. Derive a Marching Square algorithm in 2D, similar to the Marching Cube algorithm in 3D,for generating contour curves in a 2D image. Describe the details of the different cases and draw the complete case table for generating the line segments of the contour curve.

4. Algo

1) To make binary image in 2D, apply threshold ~~to the~~ ~~2D~~

1.1 Apply 1, where data value is above the isovalue.

1.2 Apply 0 where data value is below the isovalue.

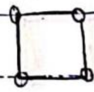
2) Threshold ~~a~~ with iso value.

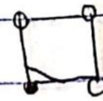
3) Convert threshold with iso value to binary image cells.

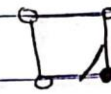
4) Give every cell a number based on which corners are true or false.

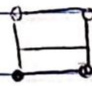
5) Use cell index to access a pre-built lookup table with 16 entries.

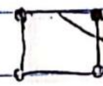
6) Apply linear interpolation between the original field data values to find exact position of the contour line along the edges of the cell

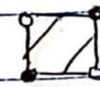
case 0 

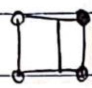
case 1 

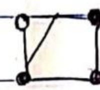
case 2 

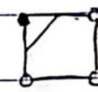
case 3 

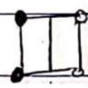
case 4 

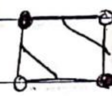
case 5 

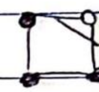
case 6 

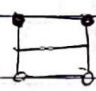
case 7 

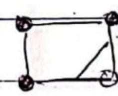
case 8 

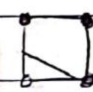
case 9 

case 10 

case 11 

case 12 

case 13 

case 14 

case 15