

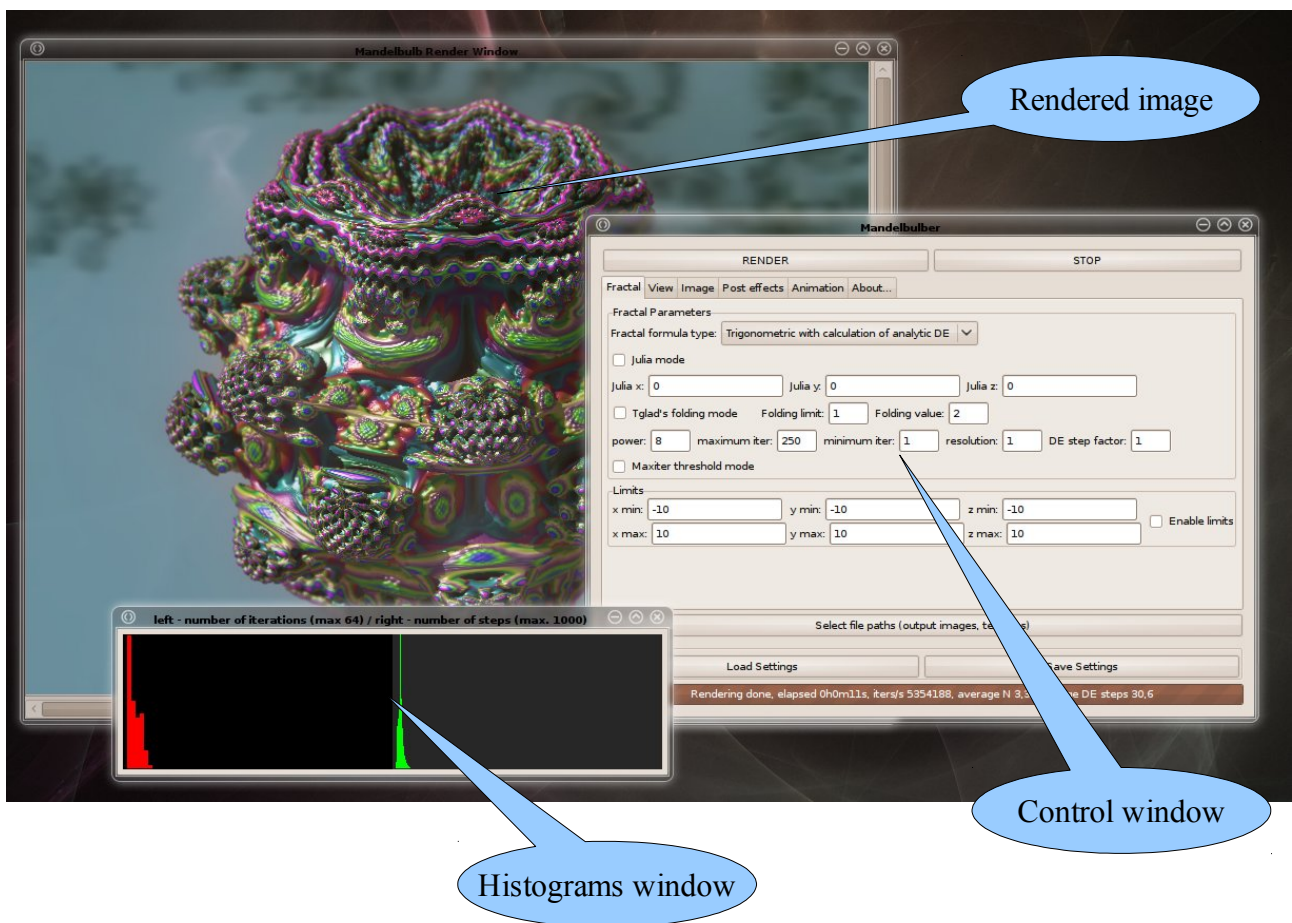
USER GUIDE



Mandelbulber

Program for rendering high quality 3D fractals

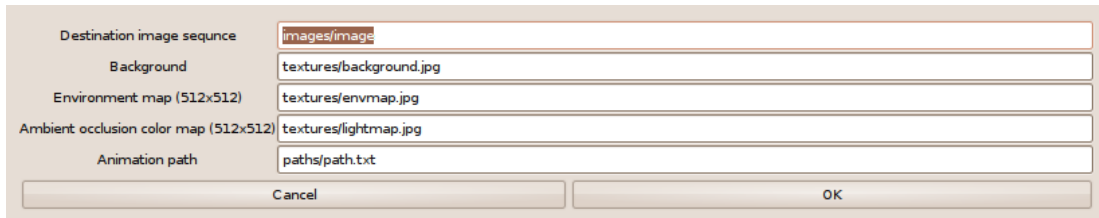
1 Main windows



2 Control window

2.1 General buttons

- **Render** – Starts image rendering. Image will be automatically saved in \$HOME/.mandelbulber/image directory
- **Stop** – Stops rendering and saves partially rendered image
- **Select file paths** – opens window with setup for paths of additional files

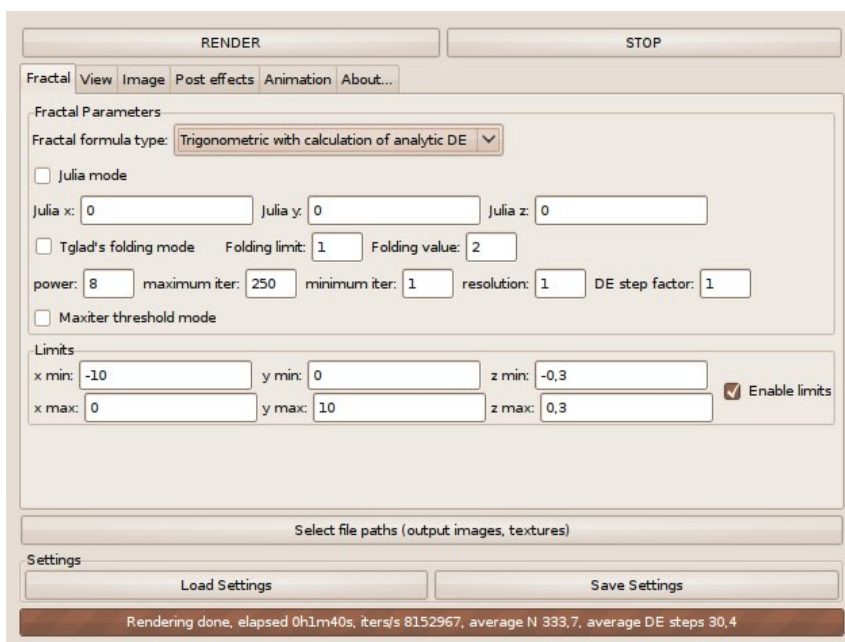


- **Destination image sequence** – path and file name prefix for saved images. To the name the program will automatically add a number to the image and a file extension of “.jpg”
 - **Background** – path of the image will be used as a image background
 - **Environment map** – path of the image will be used as a fake reflection. Resolution of this image has to be 512x512
 - **Ambient occlusion colour map** – path of the image will be used as a colour map for ambient occlusion effect. Resolution of this image has to be 512x512
 - **Animation path** – file in which will be stored the recorded flight path
- **Load setting** – load all fractal parameters from file
 - **Save settings** – save all fractal parameters to file

Settings files are text files which can be also edited using simple text editors. Important is to keep proper file syntax and names of parameters.

Default settings are stored in \$HOME/.mandelbulber/settings/default.fract file.

2.2 Fractal tab



2.2.1 Fractal parameters

- **Fractal formula type** – Selection for fractal formula type.

- **Trigonometric (Mandelbulb)**

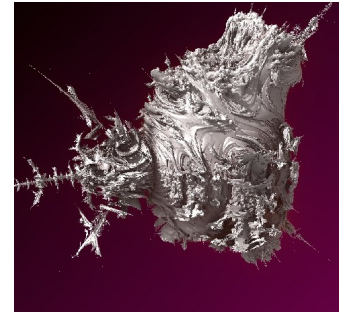
$$\begin{aligned}
 r &= \sqrt{x^2 + y^2 + z^2} \\
 \alpha &= \text{atan2}(y, x) \\
 \beta &= -\text{atan2}(z, \sqrt{x^2 + y^2}) \\
 x &= r^p \cos(p\beta) \cos(p\alpha) + a \\
 y &= r^p \cos(p\beta) \sin(p\alpha) + b \\
 z &= r^p \sin(p\beta) + c
 \end{aligned}$$



- **Trigonometric with calculation of analytic DE** – is similar to the *Trigonometric formula* but distance estimation is calculated by analytic formulas. This formula is faster than *Trigonometric* but less accurate for low power fractals and Julia fractals

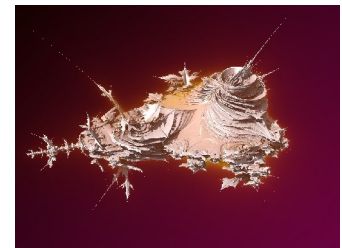
- **Polynomic power 2**

$$\begin{aligned}
 \text{newx} &= (x^2 - y^2) \left(1.0 - \frac{z^2}{x^2 + y^2} \right) \\
 \text{newy} &= 2xy \left(1.0 - \frac{z^2}{x^2 + y^2} \right) \\
 \text{newz} &= -2z \sqrt{x^2 + y^2} \\
 x &= \text{newx} + a \\
 y &= \text{newy} + b \\
 z &= \text{newz} + c
 \end{aligned}$$



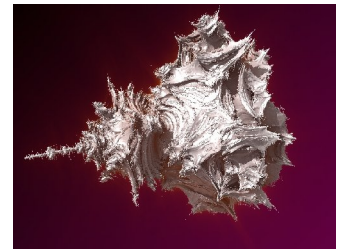
- **Polynomic power 2 – minus z**

$$\begin{aligned}
 \text{newx} &= (x^2 - y^2) \left(1.0 - \frac{z^2}{x^2 + y^2} \right) \\
 \text{newy} &= 2xy \left(1.0 - \frac{z^2}{x^2 + y^2} \right) \\
 \text{newz} &= 2z \sqrt{x^2 + y^2} \\
 x &= \text{newx} + a \\
 y &= \text{newy} + b \\
 z &= \text{newz} + c
 \end{aligned}$$



- **Hypercomplex**

$$\begin{aligned}
 \text{newx} &= x^{\gamma} - y^{\gamma} - z^{\gamma} - w^{\gamma} \\
 \text{newy} &= \gamma(xy - wz) \\
 \text{newz} &= \gamma(xz - yw) \\
 \text{neww} &= \gamma(xw - yz) \\
 x &= \text{newx} + a \\
 y &= \text{newy} + b \\
 z &= \text{newz} + c \\
 w &= \text{neww}
 \end{aligned}$$



- **Quaternion**

$$newx = x^2 - y^2 - z^2 - w^2$$

$$newy = 2xy$$

$$newz = 2xz$$

$$neww = 2xw$$

$$x = newx + a$$

$$y = newy + b$$

$$z = newz + c$$

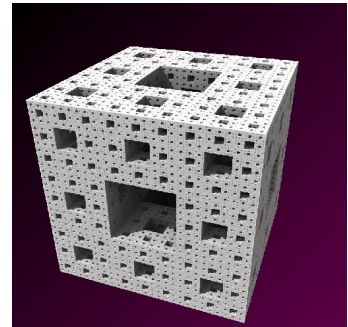
$$w = neww$$



- **Menger sponge**

details:

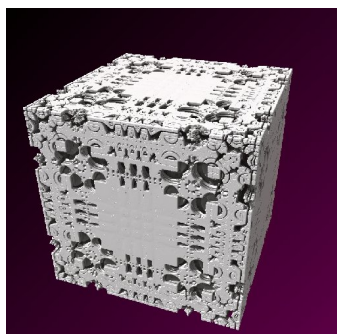
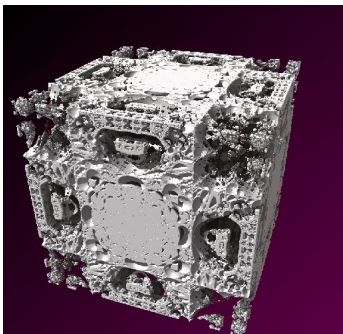
http://en.wikipedia.org/wiki/Menger_sponge



- **Tglad's formula (Mandelbox)**

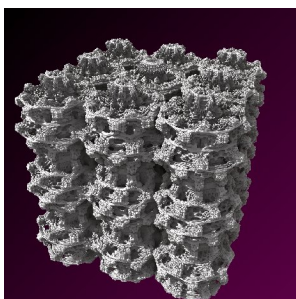
details:

<http://www.fractalforums.com/3d-fractal-generation/amazing-fractal/>



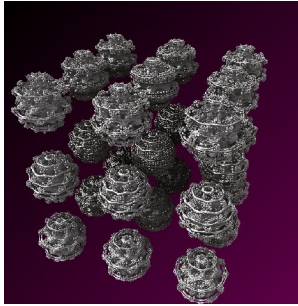
Scale parameter for Tglad's formula can be put in Power edit field (power = scale)

- **Julia mode** – enables rendering of 3D Julia fractals
- **Julia x, Julia y, Julia z** – Julia constants
- **Tglad's folding mode** – add Tglad's folding function to fractal formula
example of Mandelbulb with folding function:



(folding limit = 1, folding value = 2)

- **Folding limit, Folding value** – parameters for folding function



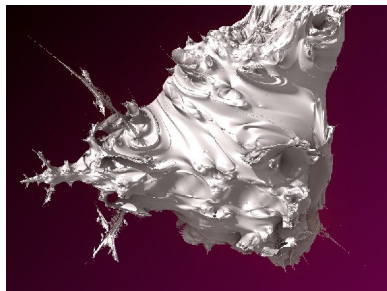
(folding limit = 2, folding value = 4)

- **Power** – power value of fractal formula:

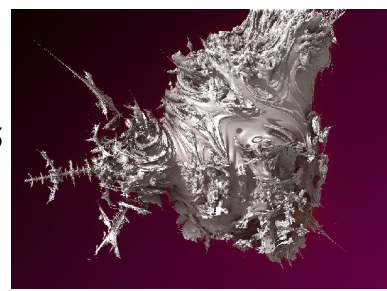
$$z_{n+1} = z_n^p + c \quad p - \text{power}$$

- **Maximum iter** – maximum number of iterations of fractal formula. Higher value gives greater amount of detail.

Max iter. = 10

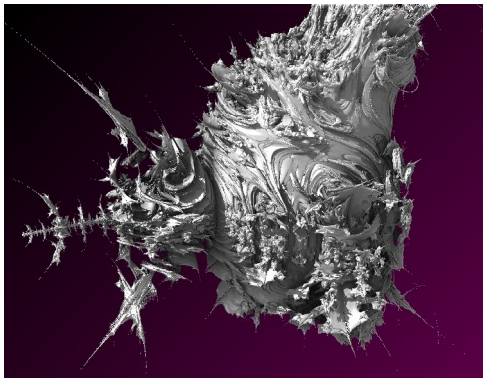


max iter. = 15

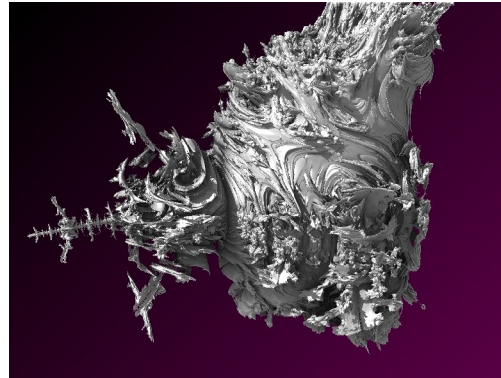


- **Minimum iter** – minimum number of iterations allows to stop ray marching. It is useful to reduce details in areas with high estimated distance and low number of iterations.

Min iter. = 1.0

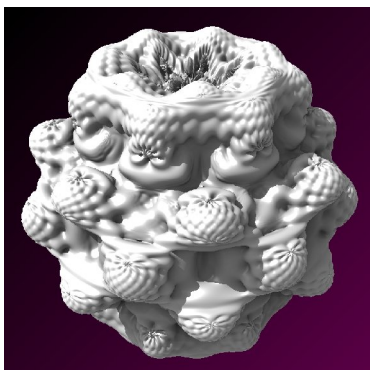


Min iter. = 10.0

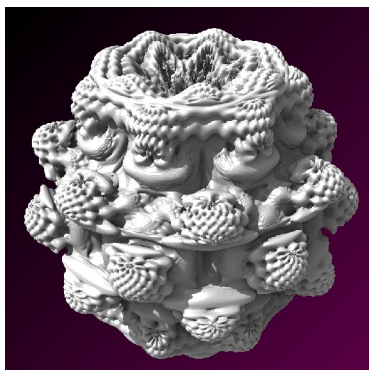


- **Resolution** – factor for dynamic threshold based on estimated distance. Higher value gives sharper details. Resolution = 1.0 is an equivalent to minimum detail size = 1 screen pixel

Resolution = 0.2



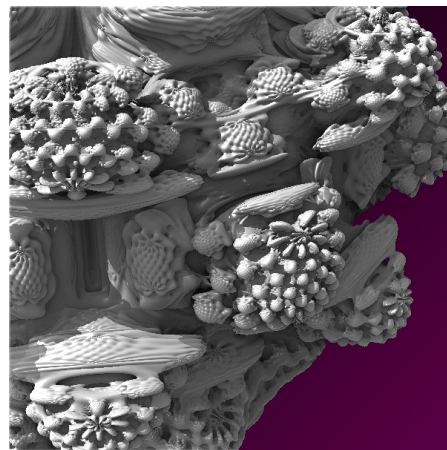
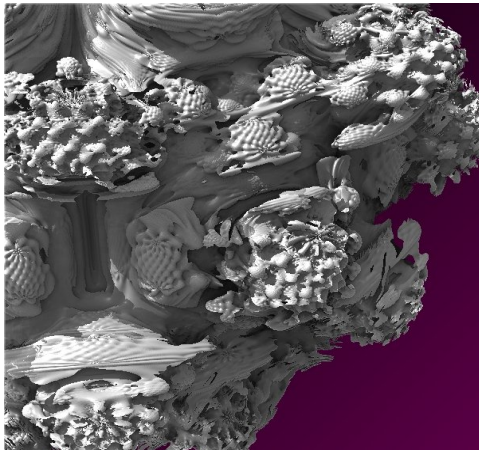
Resolution = 0.4



Resolution = 1.0

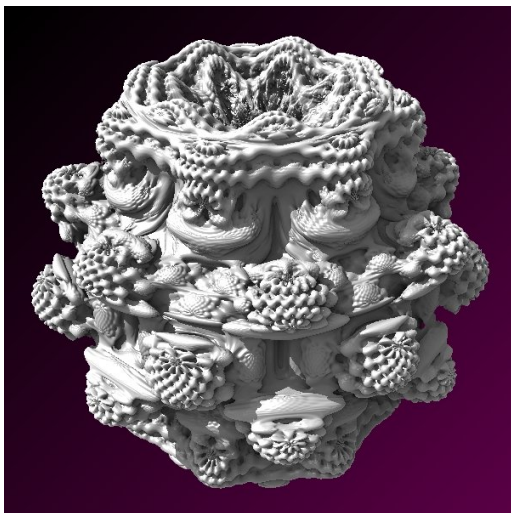


- **DE step factor** – quality of ray marching. Lower value gives greater quality (avoiding of overstepping) but slower rendering. For most fractals optimal value is between 0.5 and 1.0
to high *DE step factor* (visible errors) proper *DE step factor*



- **Maxiter threshold mode** – ray marching stops when required maximum number of iterations are achieved. Shape of the fractal is different than in distance estimation threshold mode

DE threshold mode

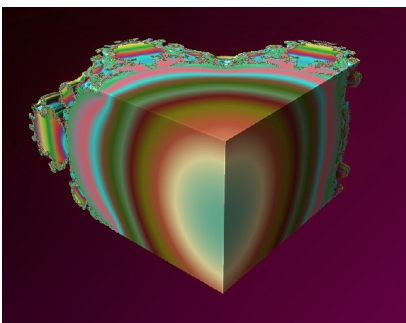


Maxiter threshold mode

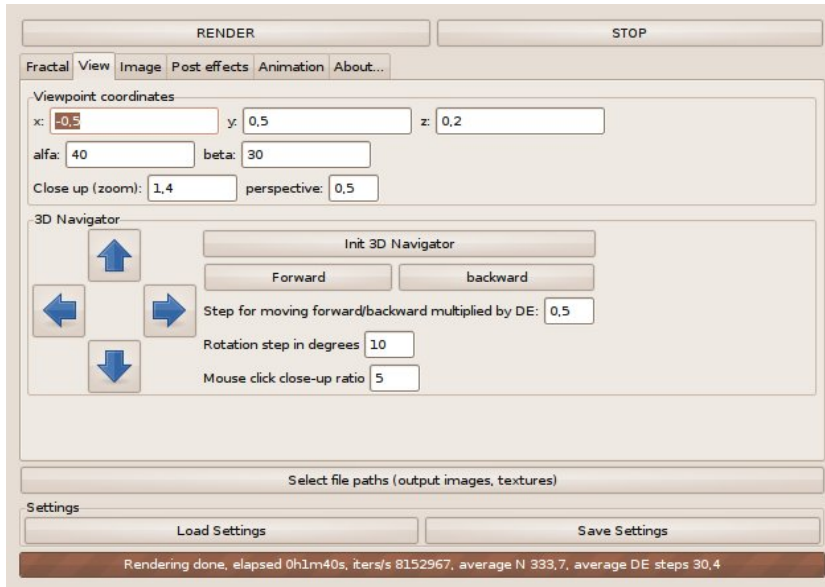


2.2.2 Limits

- **Enable limits** – fractal will be cut according to entered slicing planes
- **x_min, y_min, z_min, x_max, y_max, z_max** – coordinates of cross-section planes
Example: $x_{min}=-10, x_{max}=0, y_{min}=0, y_{max}=10, z_{min}=-0.3, z_{max}=0.3$



2.3 View tab



2.3.1 Viewpoint coordinates

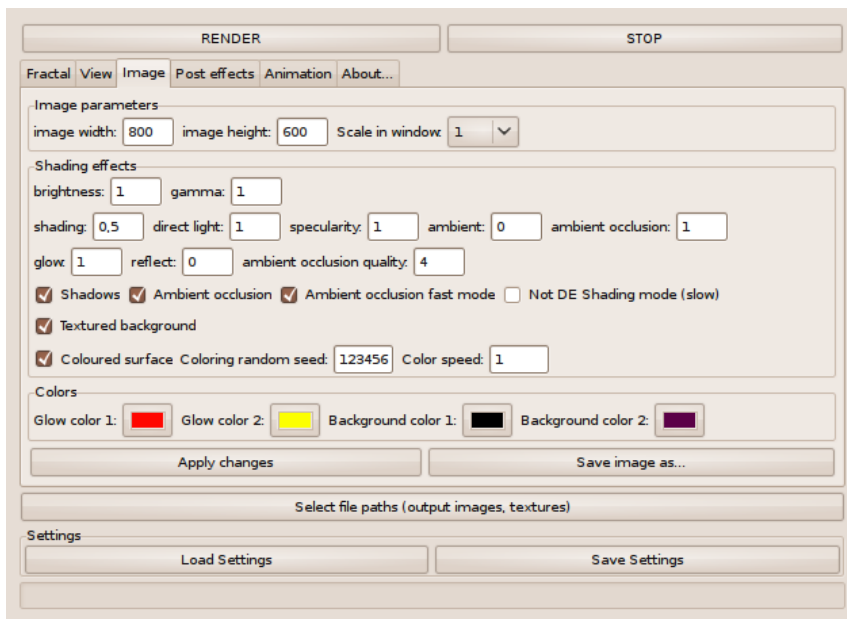
- **x, y, z** – point where the camera is looking
- **alfa** – camera rotation around vertical axis
- **beta** – camera rotation around horizontal axis
- **Close-up (zoom)** – the size of the object located on the viewpoint which will fit on the screen. Lower values takes the camera closer to the viewpoint (higher zoom). For example, Mandelbulb has a diameter of 2.2, the viewpoint is located in the centre of the fractal (0,0,0) and if the entire Mandelbulb has to be visible on screen, then the best Close-up will be around 2.5.
- **Perspective** – depth of the perspective effect. If perspective = 0, then it will be an axonometric projection.

2.3.2 3D Navigator

- **Init 3D Navigator** – moves the camera far from the object and set-up default camera angle, zoom and position
- **Forward** – moves the camera forward
- **Backward** – moves the camera backward
- **Step for moving forward/backward multiplied by DE** – camera will move on distance entered in this field multiplied by actual distance to the fractal surface.
- **Rotation step** – camera rotation step in degrees.
- **Arrows** – camera rotation
- **Mouse click close-up ratio** – after zooming by mouse clicking on the image, the *Close-up (zoom)* parameter will be divided by this value

If you indicate by mouse some interesting detail on the image and press left mouse button, the program will calculate a new camera position and moves the camera closer.

2.4 Image tab



The screenshot shows the 'Image' tab of a fractal rendering application. At the top are 'RENDER' and 'STOP' buttons. Below are tabs for 'Fractal', 'View', 'Image' (selected), 'Post effects', 'Animation', and 'About...'. The 'Image parameters' section includes input fields for 'image width' (800), 'image height' (600), and 'Scale in window' (1 with a dropdown arrow). The 'Shading effects' section contains sliders for 'brightness' (1), 'gamma' (1), 'shading' (0.5), 'direct light' (1), 'specularity' (1), 'ambient' (0), 'ambient occlusion' (1), 'glow' (1), 'reflect' (0), and 'ambient occlusion quality' (4). There are checkboxes for 'Shadows', 'Ambient occlusion', 'Ambient occlusion fast mode', 'Not DE Shading mode (slow)', 'Textured background', and 'Coloured surface'. Below these are color pickers for 'Glow color 1' (red), 'Glow color 2' (yellow), 'Background color 1' (black), and 'Background color 2' (purple). At the bottom are buttons for 'Apply changes', 'Save image as...', 'Select file paths (output images, textures)', 'Load Settings', and 'Save Settings'.

2.4.1 Image parameters

- **Image width, Image height** – size of final image
- **Scale in window** – scale of viewed image

2.4.2 Shading effects

2.4.2.1 Edit fields

- **Brightness*** – brightness of the fractal
- **Gamma*** – image gamma
- **Shading*** – intensity of the shading effect based on angle of incidence of the light



- **Direct light*** – intensity of direct light that results in casting of shadows



- **Specularity*** – intensity of specular reflection which is the mirror-like reflection of the light from a surface



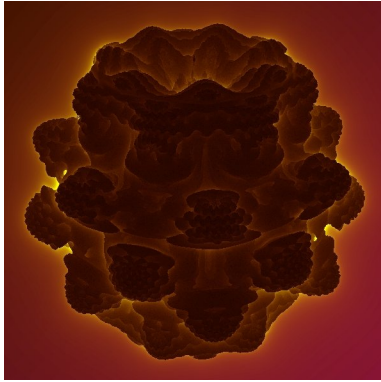
- **Ambient*** – intensity of the ambient light
without ambient light: with ambient light:



- **Ambient occlusion*** – intensity of the ambient occlusion effect (simulation of light going from all directions)



- **Glow*** - intensity of the glow effect



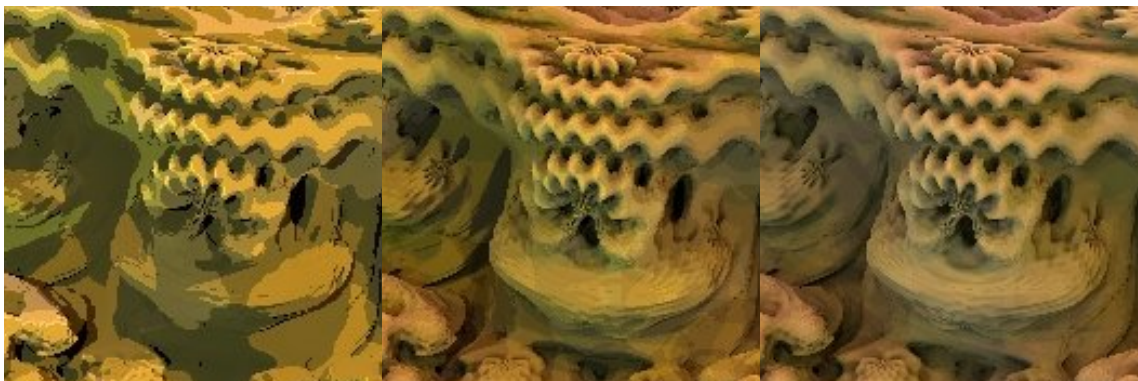
- **Reflect*** - intensity of the reflection effect. This effect based on environment mapping technique (reflections are taken from reflection map texture).
- **Ambient occlusion quality** – quality of ambient occlusion effect. This parameter determines how many rays of light will be calculated for each point.

quality	1	2	5	10	15
Number of rays	8	64	165	645	5702

quality = 1

quality = 2

quality = 3

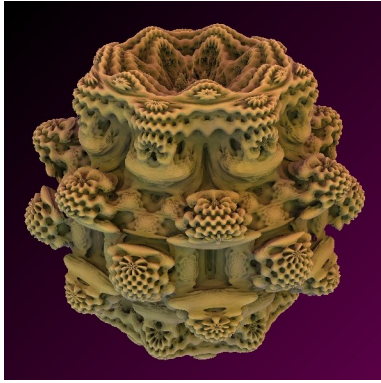


All parameters marked by * can be changed during or after rendering and updated on the image by *Apply changes* button

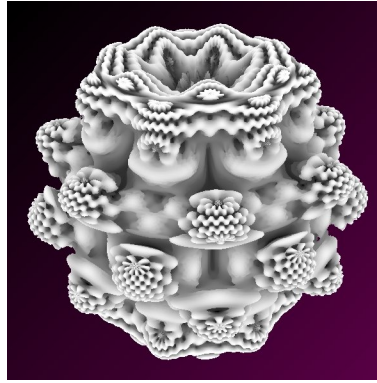
2.4.2.2 Check-boxes

- **Shadows** – enable shadows
- **Ambient occlusion** – enable ambient occlusion effect
- **Ambient occlusion fast mode** – enable rendering of fake ambient occlusion based on calculation of point orbit traps. It is very fast to render but works properly only with Mandelbulbs (trigonometric formulas) and high power value (higher than 4)

Normal ambient occlusion



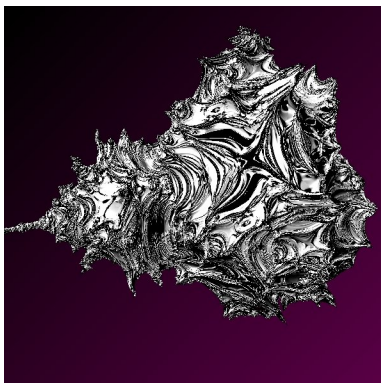
Fast mode (fake)



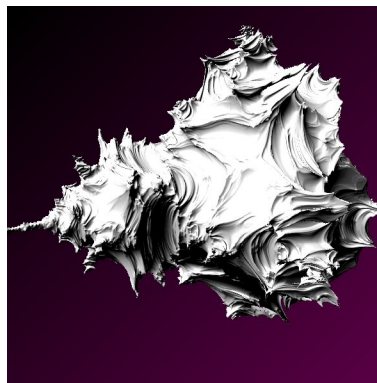
- **Not DE shading mode** – normally shading is rendered based on calculations of normal vectors from an estimated distance value. For some kinds of formulas (hypercomplex, low power mandelbulbs) it is not accurate. When *Not DE shading mode* is enabled, then normal vectors are calculated as an average gradient of binary potential function. It is very slow method but it always works.

Example for hypercomplex fractal:

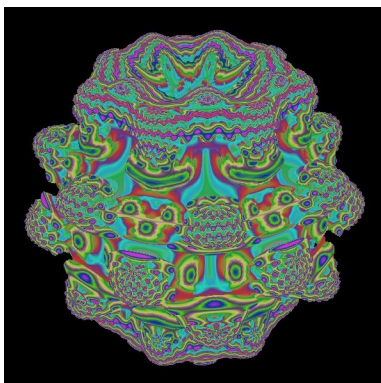
normal shading mode



not DE shading mode



- **Textured background** – background will be rendered as a textured 3D sphere
- **Coloured surface*** – enable colouring algorithm for the fractal surface



- **Random seed*** – random seed for colour palette
- **Coloring speed*** – frequency of colour variations

2.4.3 Colours

- **Glow colour 1*** – colour of glow effect in low intensity areas
- **Glow colour 2*** – colour of glow effect in high intensity areas
- **Background colour 1** – colour of background in the left top corner of image
- **Background colour 2** – colour of background in the right bottom corner of image

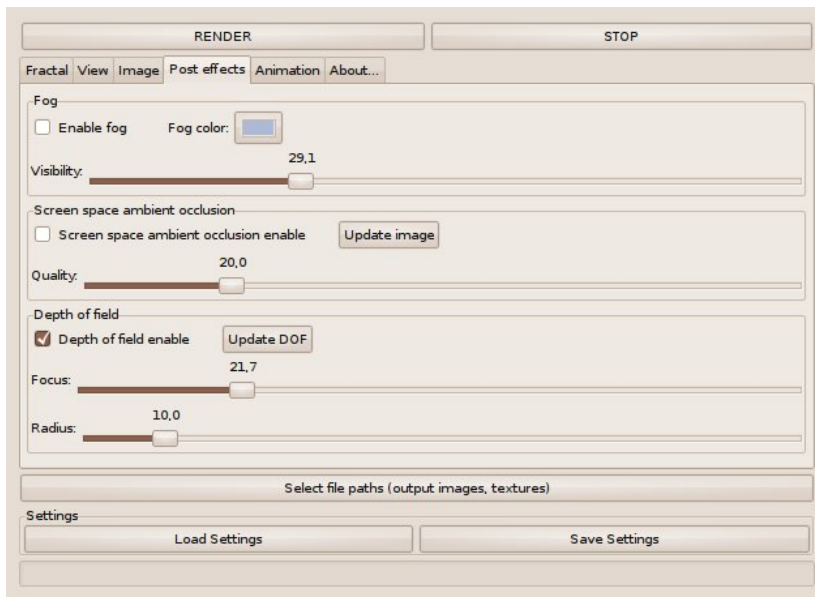
2.4.4 General

- **Apply changes** – updates image after changing shading parameters.
- **Save image as...** - saves an image in selected place.

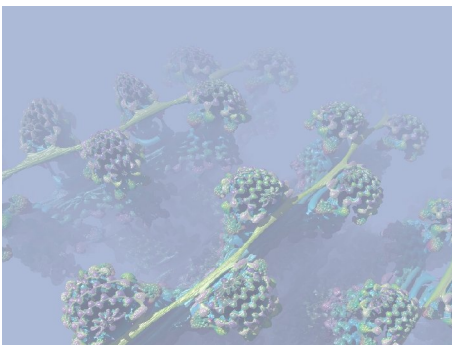
All parameters marked by * can be changed during or after rendering and updated on the image by *Apply changes* button

2.5 Post effects

These effects are based on z-Buffer and can be rendered/updated after rendering image



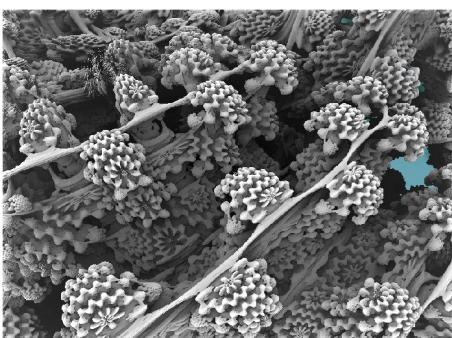
2.5.1 Fog



- **Enable** – enable fog effect
- **Fog colour** – selection of colour of the fog
- **Visibility** – visibility distance (algorithmic scale)

2.5.2 Screen space ambient occlusion

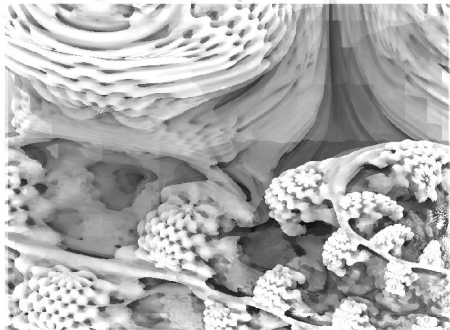
This is an equivalent of standard ambient occlusion effects. This effect is based only on depth information recorded in the screen z-Buffer. Rendering of this effect is much faster than ambient occlusion based on many rays but sometimes gives a worse result and the ambient light is monochromatic.



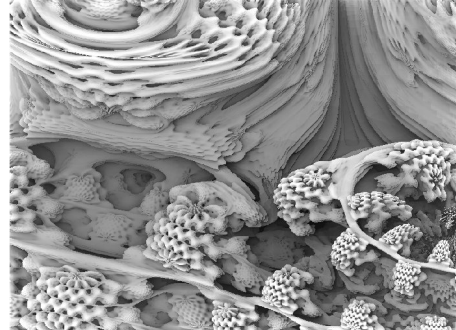
- **Enable** – enable screen space ambient occlusion effect
- **Quality** – number and resolution of fake rays. Optimum quality value is:

$$quality = \sqrt{\frac{1}{2} ImageWidth}$$

quality = 6 (image width = 800)



quality = 20 (image width = 800)



This effect has a preview during rendering with half quality

- **Update image** – update SSAO effect during and after fractal rendering

2.5.3 Depth of field

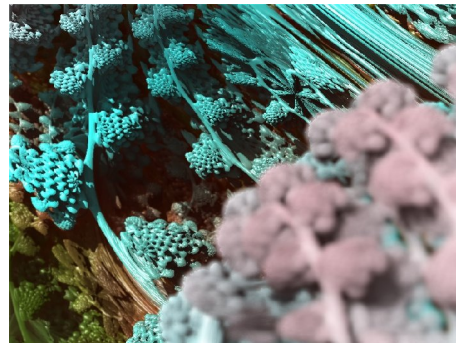
This effect simulates focus blur. Rendering of DOF is based only on depth information recorded in screen z-Buffer and can be done after fractal rendering.

- **Enable** – enables depth of field effect
- **Focus** – focal length

Low *focus* value (foreground is sharp)

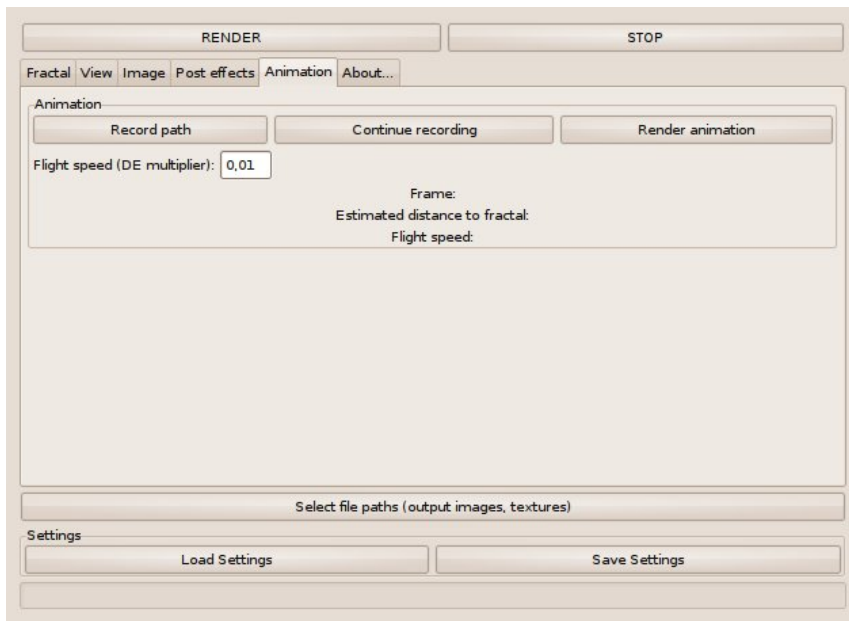


High *focus* value (background is sharp)



- **Radius** – blur radius – intensity of depth of field effect
- **Update DOF** – update deep of field effect on image

2.6 Animation Tab



- **Record path** – records flight path to the file indicated in *Select file paths function* / *Animation path*
- **Continue recording** – continues recording of flight path from last point saved in path file
- **Render animation** – render animation using file with recorded flight path
- **Flight speed** – speed of flight during recording. Resultant flight speed is a *Flight speed* multiplied by actual distance to fractal surface. It means the flight speed is automatically reduced when camera is getting closer to the fractal. *Flight speed* can be adjusted during recording by clicking left/right mouse button inside the *Render window*.

Procedure for preparing and rendering animation

1. Prepare the starting viewpoint. You can click "Init 3D Navigator", the camera will move to the default starting position. You can also put your own camera coordinates and angle, set "perspective" to around 1.2 and "Close up (zoom)" to very low value (e.g. 1e-6). You can also decide how fast the flight will be. It is adjustable by the "Flight speed (DE multiplier). Camera speed is proportional to the estimated distance to the fractal surface.
2. Set up a very low image resolution (e.g. 160x120) and "Scale in window" to higher value (e.g. 4). Since the Mandelbulber program offers high quality images instead of calculation speed, it will be faster to record flight track in low resolution.
3. Click "Record path" and navigate camera by mouse movement inside the render window. Your flight track is recorded in "paths/path.txt: file (default) and low-res stills are recorded in "images/imageXXXX.jpg" files
4. When the flight path is ready, set-up requested resolution of animation (e.g. 1280x720) and other quality parameters
5. Set up image stills destination path by *Select file paths / destination image sequence*. This folder should be empty.
6. Save actual setting by "Save settings". It will be helpful when you will want to continue rendering animation after stop.

7. Click "Render animation". Animation is now rendering according to the camera coordinates saved in path.txt file.
8. If you want to continue rendering animation you started before (e.g. after some power failure), load the proper settings file and just click on "Render animation". The program will scan which frames are already rendered and render all missed frames.

Table of contents

1 MAIN WINDOWS.....	1
2 CONTROL WINDOW.....	2
2.1 GENERAL BUTTONS.....	2
2.2 FRACTAL TAB.....	2
2.2.1 Fractal parameters.....	3
2.2.2 Limits.....	6
2.3 VIEW TAB.....	7
2.3.1 Viewpoint coordinates.....	7
2.3.2 3D Navigator.....	7
2.4 IMAGE TAB.....	8
2.4.1 Image parameters.....	8
2.4.2 Shading effects.....	8
2.4.2.1 Edit fields.....	8
2.4.2.2 Check-boxes.....	10
2.4.3 Colors.....	12
2.4.4 General.....	12
2.5 POST EFFECTS.....	13
2.5.1 Fog.....	13
2.5.2 Screen space ambient occlusion.....	13
2.5.3 Depth of field.....	14
2.6 ANIMATION TAB.....	15