Texture Builder Plugin for CamBam

[Version 1.0.1]

Purpose

Textured surfaces are commonly used in CNC machining to create interesting or contrasting backgrounds on carved items. Essentially a textured surface suitable for CNC machining is a 2.5D surface with a Z (depth) varying over an X-Y plane.

This plugin is built on the following premises:

- That the surface to be textured is a tessellation of a series of 2.5D tiles.
- Each tile can be repeated over the surface using some combination of:
  - Copying
  - Translating
  - Scaling
  - Repeating on an X-Y grid, or around a circular arc in the X-Y plane.
- The tile element must be predefined (using some other tool) as:
  - a height cloud (a set of X,Y,Z coordinate points) in a CSV file,
  - an STL model (Sterolithographic file, in ASCII or Binary formats),
  - a RAW file (sets of X,Y,Z point triplets defining each surface triangular surface patch, as defined for CamBam, in ASCII format), or
  - an image file (BMP. GIF, JPG, PNG or TIFF formatted) where the grey scale values are to be interpreted as a height map (in the range 0 to 255).
- Once the scene is constructed, the complete scene surface can be saved as a XYZ height cloud, an STL file or a RAW file, for input into CamBam, or other CAM modellers.

Related Tools and Potential Contributions

To build a tile element to form the required texture various support tools can be used to help, each performing a particular task in the process. The following tools are the free ones, or at least have a free version; there may be others also. Each has its own strengths and weaknesses and users may find any particular one more or less useful for their specific texture building requirements.

Art of Illusion (artofillusion.org**):**

- Creates meshes from scratch
- Edits/simplifies meshes
- Manages various file formats
- Moderate learning curve

L3DT – Large 3D Terrain (www.bundysoft.com**):**

- Specifically aimed at producing terrain models.
- Creates meshes from scratch
- Edits meshes
- Manages various file formats
- Gentle learning curve, but limited surface styles

MeshLab (meshlab.soureforge.net**):**

- Edits meshes

- Manages various file formats
- Gentle learning curve

Sculptris (pixologic.com**)**

- Create meshes from scratch
- Edits meshes
- OBJ files only
- Gentle learning curve, limited complexity

Blender (blender.org)

- Creates meshes from scratch
- Edits meshes
- Manages various file formats
- Steep learning curve

GeoGen (geogen.cz)

- Creates bitmap image height maps parametrically.
- Quite powerful scripting language.
- Includes a range of tools for generating "natural" surfaces.
- Produces bitmap-based height maps.
- Requires programming skills.

FreeCAD (www.freecadweb.org))

- A general CAD drawing/modelling package
- Can create 3-D models and export meshes.
- Limited mesh editing.
- Converts various file formats.
- Scripting possibilities.

OpenSCAD (www.openscad.org)

- 3D modelling using parametric constructive solid geometry
- Very powerful scripting language for model building
- Requires programming skills
- Exports to STL
- Moderate learning curve.
- Model building is fast but rendering (to get the final accurate surface definition) is slow!

OpenJSCAD (openjscad.org)

- 3D modelling using parametric CSG
- Quite similar to OpenSCAD, but uses Javascript
- Can be run within a browser or as a standalone package, or using a command line interface.
- Exports to STL
- Requires programming skills.
- Moderate learning curve

ShapeSmith (www.shapesmith.net)

- A browser-based 3-D modelling tool
- Has some parameter-based GSG capabilities using variable to define dimensions and positions.
- Can export STL files

Tinkercad (www.tinkercad.com)

- A browser-based 3-D modelling tool
- Reasonable selection of components
- Can export STL files.

Some suggestions (my preferences):

- For mesh construction, simplification and general editing – Art Of Illusion
- For accurate 3-D parametric modelling building – OpenJSCAD or GeoGen

File/Model Formats and Conversions

Various combinations of tile model and file types are used depending on input and required output file formats. Internally, the plugin maintains two model formats for tiles:

- Cloud – where the tile is represented by a collection of X,Y,Z coordinates.
- Face – where the tile is represented by a collection of triangular faces.

Depending on the data available in the source file the internal model formats are retained until a Save operation is required, then the internal formats are converted to the required output file format as shown in the following table:

Input file format	Internal format	Save file options	Import into CamBam
Image (bmp, gif, jpg, png, tiff)	Cloud	XYZ	No
		STL, RAW	Yes
XYZ (csv)	Cloud	XYZ	No
		STL, RAW	Yes
RAW	Face	XYZ	No
		STL, RAW	Yes
STL	Face	XYZ	No
		STL, RAW	Yes

Different tile formats can exist together in one model for manipulation in the plugin, though a single specified format is used when the model is saved.

Note that while a XYZ file can be saved, it cannot be imported directly in to CamBam. The plugin can, of course, be used to convert a CSV file into a suitable STL or RAW file that can be imported into CamBam.

Installation

The TextureBuilder.dll file needs to be copied into the CamBam Plugins folder, and CamBam restarted. The plugin will then appear in the Plugins menu.

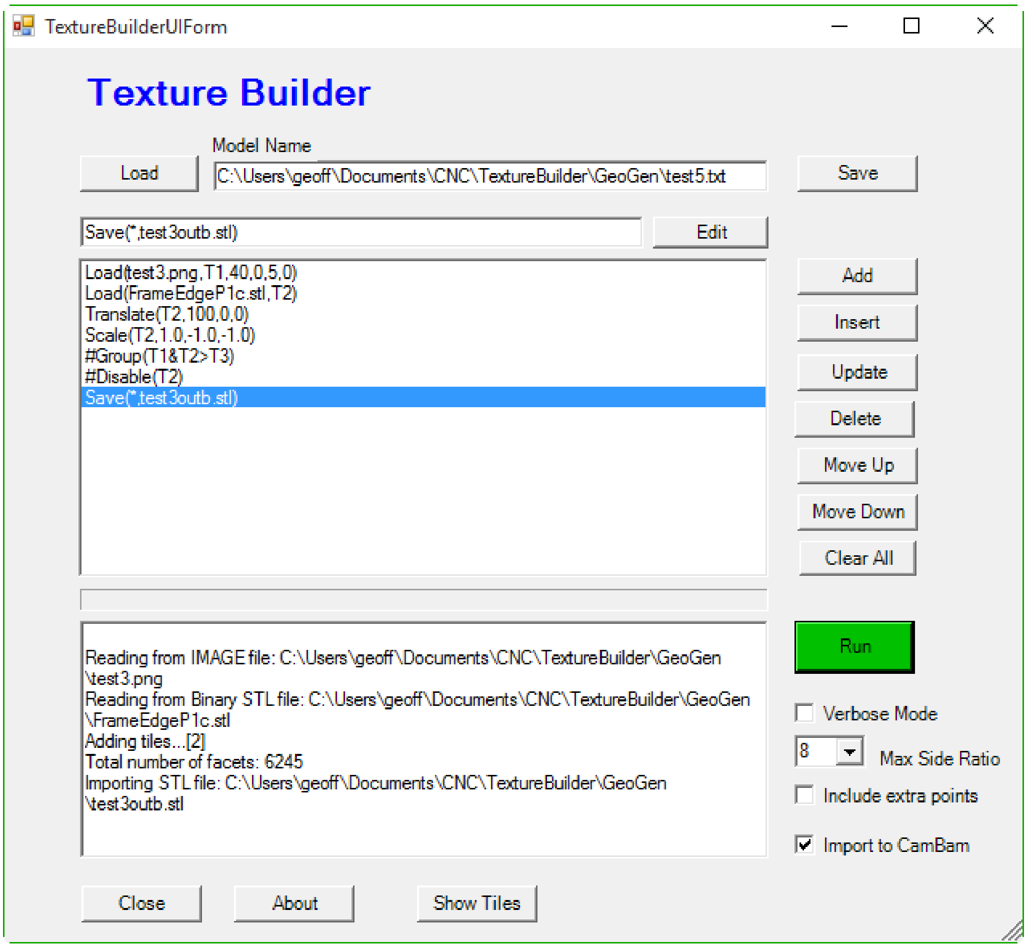
Operation

The plugin operates generally as follows:

1. Create a tile model using another tool.
2. Import the tile definition (one or more).
3. Manipulate the tiles (copy, scale, translate, repeat, etc. as required)
4. Create a combined surface model.
5. Save this result in a file.
6. Optionally import the model into CamBam for further processing and cutting path generation.

The modelling process is described by a script that performs each step in sequence.

The Texture Builder UI is launched from the plugins menu:



Model Name: C:\Users\geoff\Documents\CNC\TextureBuilder\GeoGen\test5.txt

Save(*,test3outb.stl)

```
Load(test3.png,T1,40,0,5,0)
Load(FrameEdgeP1c.stl,T2)
Translate(T2,100,0,0)
Scale(T2,1.0,-1.0,-1.0)
#Group(T1&T2>T3)
#Disable(T2)
Save(*,test3outb.stl)
```

```
Reading from IMAGE file: C:\Users\geoff\Documents\CNC\TextureBuilder\GeoGen
\test3.png
Reading from Binary STL file: C:\Users\geoff\Documents\CNC\TextureBuilder\GeoGen
\FrameEdgeP1c.stl
Adding tiles...[2]
Total number of facets: 6245
Importing STL file: C:\Users\geoff\Documents\CNC\TextureBuilder\GeoGen
\test3outb.stl
```

☐ Verbose Mode

8 Max Side Ratio

☐ Include extra points

☑ Import to CamBam

The items in this dialog are:

1. The <Load> button will load the script from a ".txt" file.
2. The <Save> button will save the currently defined script as a ".txt" file.
3. The loaded script appears in the <Script> field.
4. To create a script command:
    a. Type the command into the second from top text box (the <Editor> text box)
    b. Click the <Add> button.
    c. The command is appended to the script in the <Script> field.
5. To edit a command:
    a. Click in the required command line in the <Script> field.
    b. This command is placed in the <Editor> text box
    c. Edit the command
    d. Click the <Update> button
    e. The edited command replaces the original
6. To insert a new command before an existing command:
    a. Click on the command that will come after the new command
    b. Type the new command into the <Editor> text box.
    c. Click the <Insert> button.
    d. The new command is inserted before the selected command.
7. To delete a command:
    a. Select the required command.
    b. Click the <Delete> button
    c. The selected command is deleted.
8. To move a command up in the script sequence:
    a. Click on the required command.
    b. Click the <Move Up> button until the command is in the required position.
9. To move a command down in the script sequence:
    a. Click on the required command.
    b. Click on the <Move Down> button until the command is in the required position
10. To clear all commands from the script:
    a. Click on the <Clear All> button (with care!)
11. The <Run> button will execute the script:
    a. The progress will be shown in the progress bar (some scripts take a little while!)
    b. Some progress explanations are displayed in the lower <Progress> text box.
    c. If the <Verbose Mode> option is selected, prior to running, then more progress information is displayed.
    d. If the <Import to CamBam> option is chosen the resulting model is automatically imported into CamBam (STL and RAW model files only) after it is saved.
    e. When saving a model to a XYZ file from a tile that is internally in a Face structure then there is an option to add additional points to the X,Y,Z point cloud for those surface triangles that are rather thin. Some Face models may contain sets of rather thin triangles (these may quite accurately represent the model shape). It may be useful to include some intermediate points in the output point cloud (to get a more uniform point coverage). The <Include extra points> checkbox will enable this option. The <Max Side Ratio> combo selects the side ratio for those surface triangles that will have some intermediate points added along their longer edges. If the length of an edge is more that the set ratio times the length of the shortest edge, then additional points are added to the output cloud file (experimental!).
12. The <Show Tiles> button provides a list of generated tiles (see later)

The commands available are (when an alternative language is available the commands may appear in that language):

Command	**Short Form**	**Description**	**Examples**

Load	L	Loads a tile file and creates a new tile in the model	Load(file.stl,T1)
			Load(file.png,T1,20,0.0,0.0,10.0)

Save	S	Saves a tile, or the whole model; the file format depends on file extension specified (csv, stl, raw)	Save(T1,file.csv)
			Save(*,file.raw)

Translate	T	Translates a tile by a specified (X,Y,Z) distance	Translate(T1,Xt,Yt,Zt)
Rotate	R	Rotates a tile by a nominated angle about a nominated X,Y location in the X-Y plane	Rotate(T1,Angle, Xo,Yo)
Scale	SC	Scales a tile for X,Y,Z scale ratios with reference the scene origin.	Scale(T1,Xs,Ys,Zs)

Copy	C	Copies a tile to a new tile, offset by X,Y,Z from the current location	Copy(T1,Xt,Yt,Zt)
			Copy(T1>T2,Xt,Yt,Zt)

RepeatLinear	RL	Duplicates a tile a number of-times over an X-Y grid	RepeatLinear(T1,Nx,X,Ny,Y)
RepeatPolar	RP	Duplicates a tile a number of times around a circular arc centred at X,Y	RepeatPolar(T1,Na,A,Xo,Yo)

Group	G	Group the selected tiles into a single tile, the sources tiles are not removed, but are disabled. Only tiles of the same internal format (Cloud or Face) can be grouped.	Group(T1&T2&T3>TT)
			Group(*>TTG)

Enable	E	Enables a tile for use in modelling. Tiles are enabled by default.	Enable(T1)
Disable	D	Disables a tile for use in modelling. A disabled tile will not be used in modelling.	Disable(T1)
Rename	RN	Renames a selected tile	Rename(T1>A1)

The command syntax is like this:

```
    <command>(<arg1>, <arg2>, …)
```

The full command names are case sensitive, the short forms are not.

Tile names are user defined literals (any sequence of characters, no quotes are used). Tile names can be composed of:

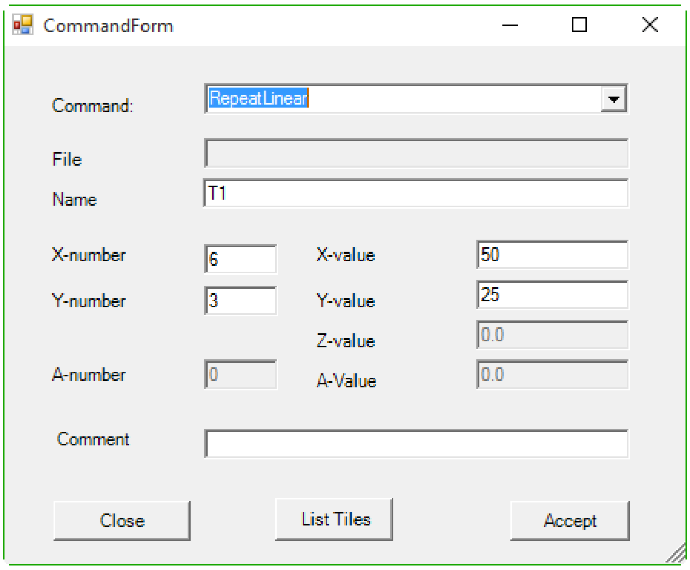
- Any sequence of letters or digits, but
- Must not include any of: ‘,’, ‘&’, ‘>’, ‘(‘, ‘)’,’+’,’-‘

For example, the script:

Load(CrossA.stl,CC)	Loads tile from the file “CrossA.stl” from the current working directory and creates a tile named “CC”.
RepeatLinear(CC,5,10,3,15)	Repeats the tile “CC” over a rectangular array, 5 times in the X-direction at intervals of 10 units, and 3 times in the Y-direction at intervals of 15 units.
Save(*,Cross.stl)	Saves all (enabled) tiles in the model to an STL model file named “Cross.stl” in the current working directory.

The wild card tile name (“*”) can be used to select all tiles in the current scene, where appropriate. A command beginning with “#” is taken to be a comment and is not executed.

Top assist with specifying commands the <Edit> button can be used to launch the command editor, like this:



Command: RepeatLinear

File

Name T1

X-number 6 X-value 50

Y-number 3 Y-value 25

Z-value 0.0

A-number 0 A-Value 0.0

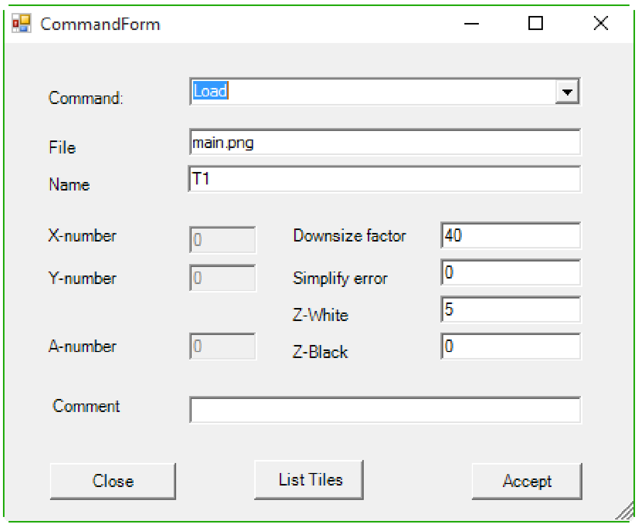
Comment

In this dialog the various arguments, only those that are valid for the currently selected command, are enabled and ready to be entered or updated. On <Accept> the fully formatted command is placed into the <Edit> text box on the main dialog. The <Add>, <Update> or <Insert> should then be used to insert the command into the script.

The fields are:

- <Command> combo: to choose the required command.
- <File> text field: the name of the tile file (in the currently active user directory).
- <Name> text field: the name of the nominated tile. Note that in the <Name> field:
  - there will be a single name for a tile (or *, for all active tiles), or
  - the names of one, or more source tile names separated by '&' characters, then a '>' character before the destination tile name.
- <Xnumber>, <Ynumber> and <Anumber> text fields: the number of repeats in X, Y and Angle directions as required.
- <Xvalue>, <Yvalue>, <Zvalue> and <Avalue> text fields: the required X, Y, Z and Angle values.
- <Comment> text field: the text for a comment command.

For the Load command, and if the file to be loaded is an image, then the following dialog variation appears:



Command: Load

File main.png

Name T1

X-number 0

Y-number 0

A-number 0

Downsize factor 40

Simplify error 0

Z-White 5

Z-Black 0

Comment

In this case, and for image files only:

- <Downsize factor> text field: a factor to scale down the imported image, e.g. a value of 10 will reduce a 1000x1000 pixel image to a 100x100 pixel image using a simple average of adjacent pixels. This step creates a new (complete) grid of X,Y points and with Z values computed from the RBG values at each pixel..
- <Simplify error> text field: The maximum surface error for triangle patch simplification. A point in the bitmap will be removed if the difference of its height value and the average of the four adjacent pixels is less that this error value. A value of 0.0 will prevent any surface simplification. The error value is on the 0-255 scale for pixel grey values.
- <Z-White> text field: The Z value to which a pixel value of 255 is mapped to.
- <Z-Black> text field: The Z-value to which a pixel value of 0 is mapped to. Setting <Z-Black> greater than <Z-White) will, in effect, invert the height map.

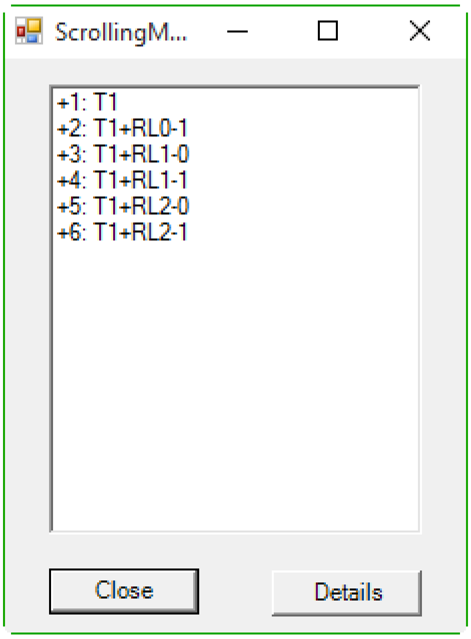
These fields will be ignored for imports from non-image files (i.e. XYZ, RAW and STL).

Some things to note about image processing:

- Large images do need to be down sized to produce manageable tile models. Some experimentation is required to suit individual needs.
- The size of the resulting tile model is set by the units being used in CamBam, e.g. an image of 1000x1000 pixels, down sized by a factor of 20 will produce a tile of size 50x50 pixels with each pixel being taken as single measurement unit (e.g. mm or inch). Further scaling will possibly be required to get a tile of suitable size for CamBam in the appropriate units.
- Surface simplification may be useful if the tile surface has significant flat regions. Experimentation is required to see the effects, and any benefits, the simplification process may offer.
- Any computed Z values outside the range Z-Black to Z-White will be given Z-Black or Z-White values as appropriate.
- The pixel-value to Z-value is a linear mapping.
- The grey scale value is computed from the pixel RGB value using:

```
zValue = 0.3*pixel.R + 0.59*pixel.G + 0.11*pixel.B
```

When tiles are repeated, new names are generated to uniquely define each tile. The names generally describe what operation has been performed on the original tile. Sometimes you may need to refer to one of these tiles by name. The <Show Tiles> button pops up a full list of tiles generated by the script, like this:

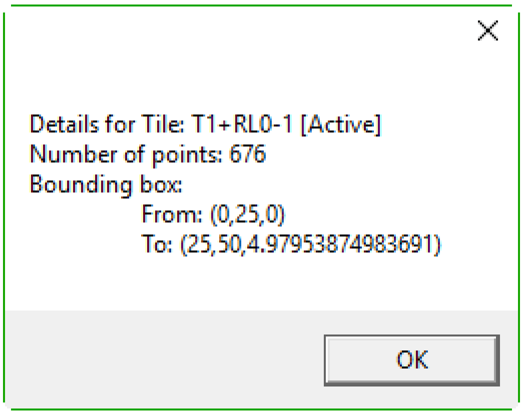


```
+1: T1
+2: T1+RL0-1
+3: T1+RL1-0
+4: T1+RL1-1
+5: T1+RL2-0
+6: T1+RL2-1
```

Tiles marked with a ‘+’ are enabled, those marked with a ’-‘ are disabled.

Generated tiles are only known about once the script has executed, so it might be useful to execute a partially complete script, especially one with no Save operation (that can take some time for larger models), to obtain a list of generated tile names. Specific tiles can then be selected for further operations (including deletion, or enabling/disabling)

The <Details> button provides a brief summary for the currently selected tile:



Details for Tile: T1+RL0-1 [Active]
Number of points: 676
Bounding box:
From: (0,25,0)
To: (25,50,4.97953874983691)

Feedback

Please send any feedback or bug reports to geoff AT cadplan.com.au, or post a response on the CamBam Forum.

Some General Tips

1. A tile model (2.5D) should not contain any vertical cliffs or undercuts as this may (will?) cause the triangulation algorithm to fail.
2. Tiles that are often required to fit neatly together with no edge effects. Edges should be-matched (top/bottom and left/right) to avoid visible “seams” between tiles (though this may be a required “feature”).
3. Complex tiles may need to be simplified (to reduce number of surface patches) to reduce computational times and model sizes.
4. If a Texture Builder script seems to just stop (hang), it is probably a result of a command error, including misspelt tile names.
5. Images to be used to create tile elements may need to be pre-processed (by other image processing packages) to provide the best possible height maps based on their RGB/grey scale pixel values.

Texture Builder Plugin Versions

Version	Date	Notes
1.0.1	21/11/2015	First version for feedback and comment.