DeskArtes 3Data Expert 16.0

Advanced Color Model Handling

First Edition

"3Data Expert 16.0: Advanced Color Model Handling".

February 2026

Contact Address

DESKARTES OY

Takomotie 8

FIN-00380 HELSINKI

Finland

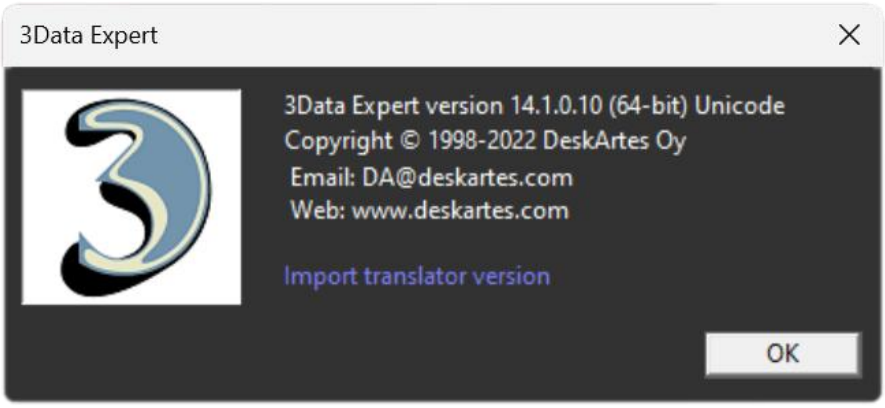
https://www.deskartes.com

Foreword

The *3Data Expert 16.0: Advanced Color Model Handling* document provides an introduction to the advanced 3D color model handling with 3Data Expert software: creating solid model from bad input data with Solidify command (aka “shrink wrap”), converting color per vertex models to textured models for reduction and Texture painting for color 3D Printing.

The target audience of this document is users who have acquired either the *Base* or the *Color* module of 3Data Expert.

You can check your version with Help > About DeskArtes 3Data Expert command:



Example 1 – Fully Automatic Repair with Solidify

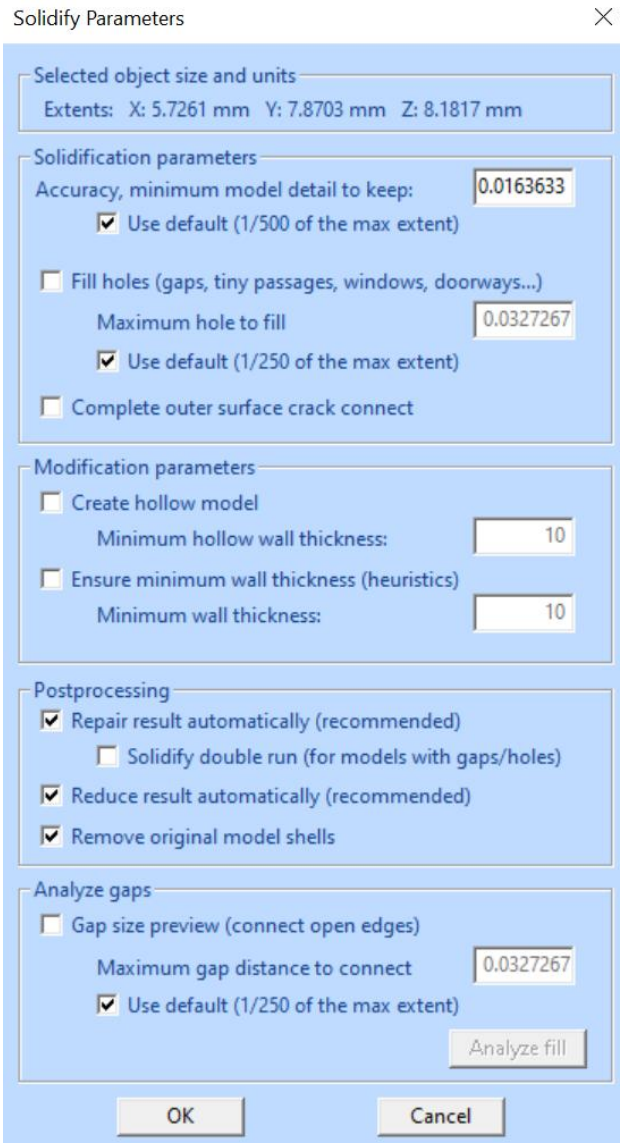
With this example we will introduce the Solidify command and perform a fully automatic repair for a color model with multiple shells, overlapping surfaces and intersecting surfaces.

In this example the Modify Faceted > Solidify command is used to fix the model for 3D Color Printing. The new command parameter dialog is as follows:

Accuracy parameter defines the minimum detail to reproduce in the repaired result. The model space is subdivided into voxels according to the given Accuracy and geometry conflicts are solved one by one in each voxel.

Fill holes function and ***Maximum gap to fill*** parameter causes the command to fill small holes in the model. To be able to separate the inside from the outside and to provide a correct result model, gaps must be closed with *Fill holes* function during the Solidify command.

Create hollow model will generate a hollow model with the given ***Minimum hollow wall thickness*** during the Solidify command. This function is at a beta stage.



Ensure minimum wall thickness will generate a thickness to open surfaces in the model. This may be necessary if not all holes in the model can be filled but a view inside the model is needed (through a window or similar hole). This function is at a beta stage.

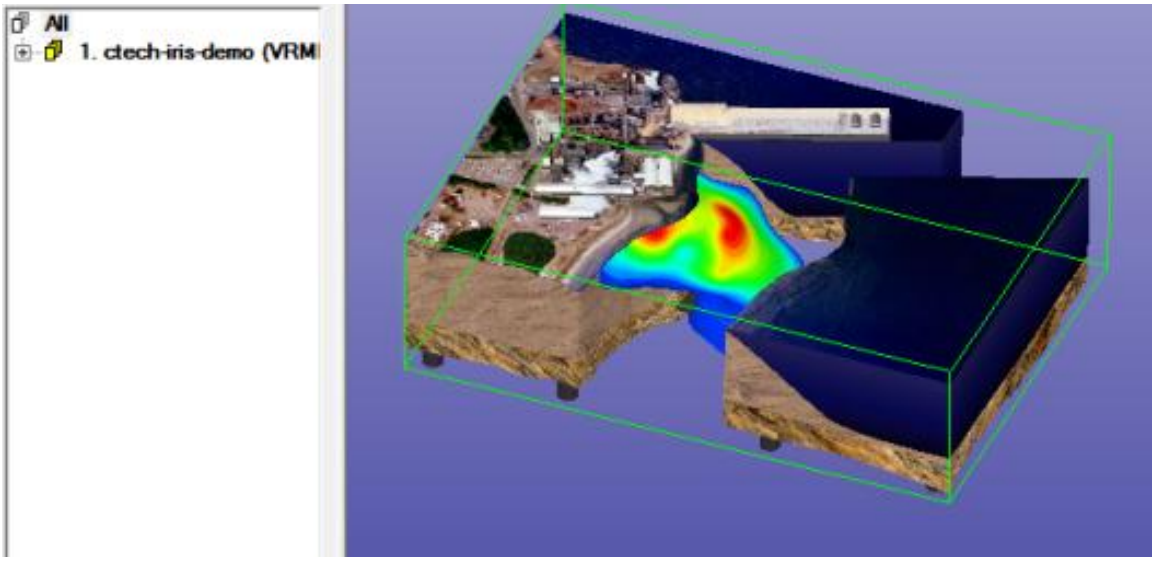
Repair result automatically will close all remaining gaps in the model after the actual Solidify operation. This is recommended although it can be run later too.

Reduce result automatically will run the Reduce command on the data. If not reduced, the Solidify command will produce typically millions of triangles. The reduce goal is to reduce triangles until the edge lengths are similar to the average edge length of the original model.

Note: Fill holes may add extra structures in the narrow passes in the model. You should always try to use as small ***Maximum gap fill*** *value as possible. Sometimes it might be better to close the largest holes with Triangle Edit before running Solidify command.*

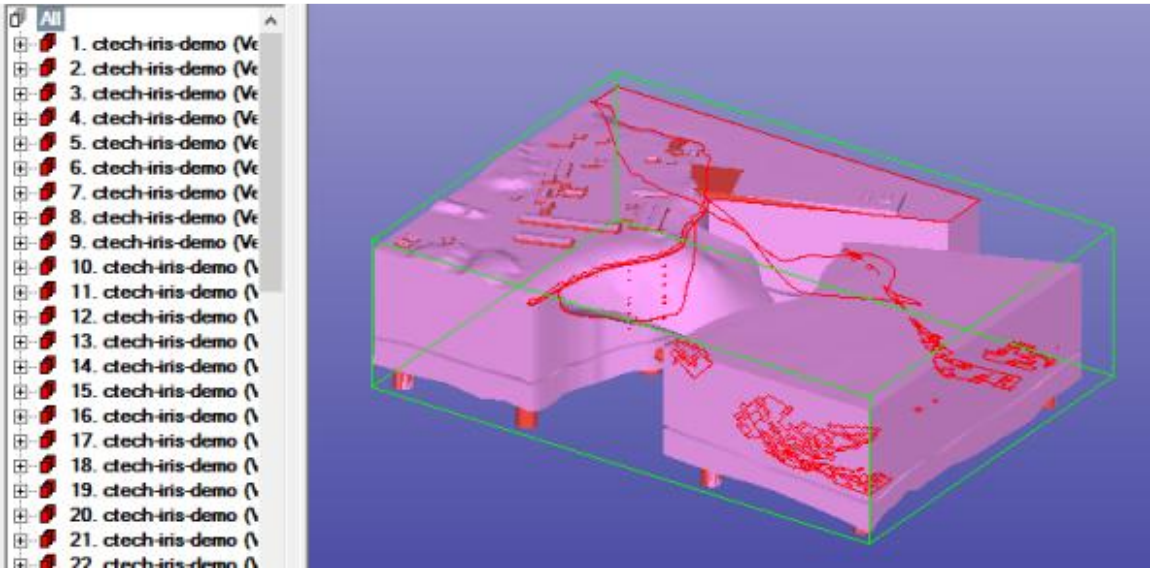
Running Solidify on geological model

The example file is a 3D model generated from a geological data set (*ctech-iris-demo.wrl*). The example model is a fully colored model with both textured surfaces and color per vertex surfaces:

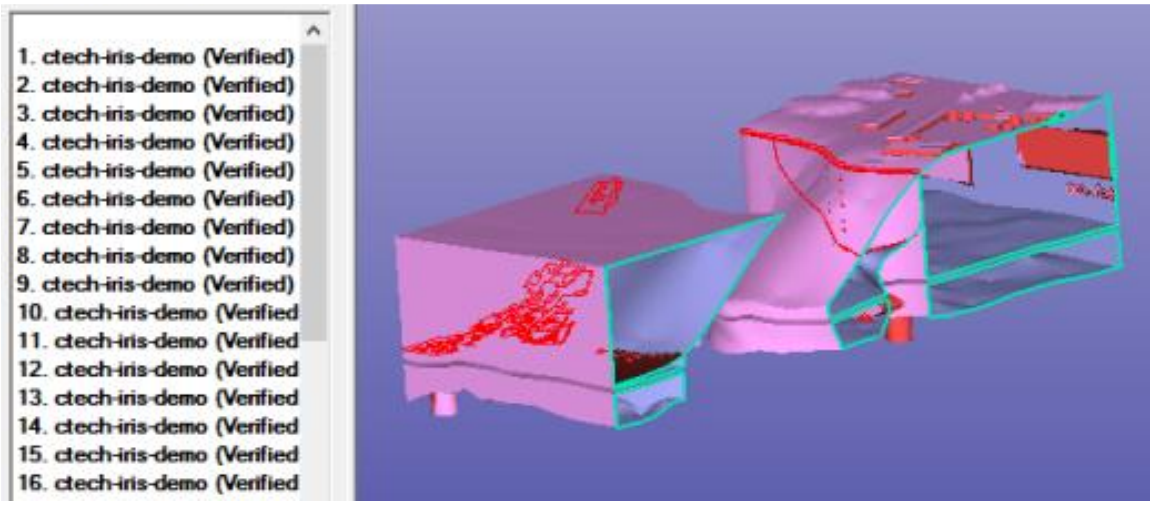


Although the model can be run through the Solidify command without any other preparation, we will run the model through Fix Model > Verify Shells command. This will show us all the errors as well as help us to determine if there are large gaps which need to be closed prior to the command.

After input and verification, the model shows several errors which are difficult or impossible to fix with normal topology generation or Boolean commands. There are several separate shells, open surfaces as well as overlapping and intersecting shells.



Some more investigation to the model shows the overlapping shells. Also, it reveals that the outer surfaces in the model will generate four separate closed areas. Now we do not need to use the Fill hole command to close gaps in the model and thus avoid generating extra structures in the result.

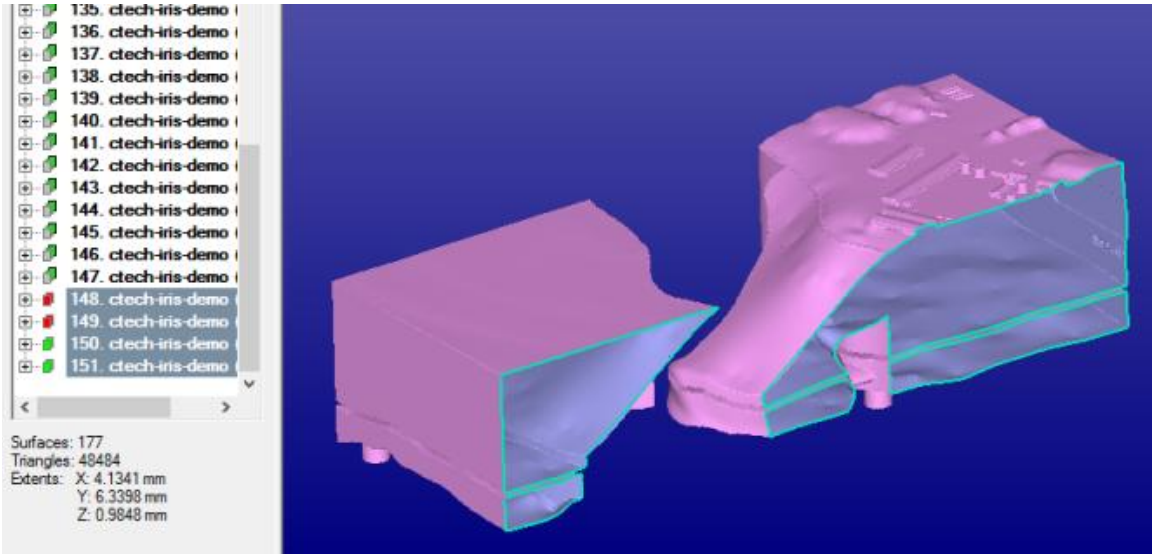


When ready investigating the model we can run the Solidify command with the default values given when the command is started:

- **Select** Model Tree root
- Give the **command Fix Model > Solidify**
- **Press OK** to start the command

Solidify command will split the model space in voxels according to the given *Accuracy* parameter. With ctech-iris-demo.wrl model this will generate up to 500x360x135 = 24.3 million voxels. Intensive calculations are performed on the voxel data and this will take several minutes to finish (solidify, repair, reduce). The processing will take about 3 GB RAM. Generally, we recommend at least 16-32 GB main memory when using the Solidify command.

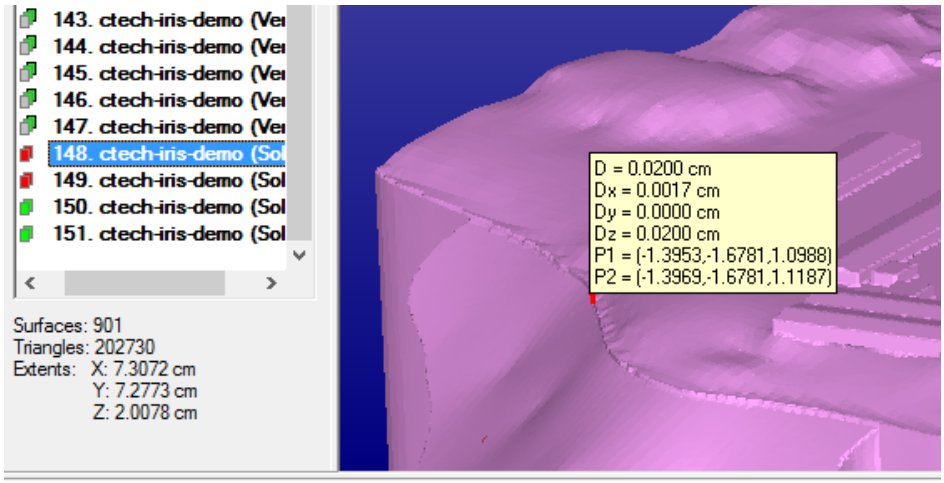
The result is 4 separate shells in the end of the Model Tree. Each shell is now a topologically correct shell ready for 3D printing. There may be minor triangle-triangle intersections left in the data, but these should not prevent printing the model.



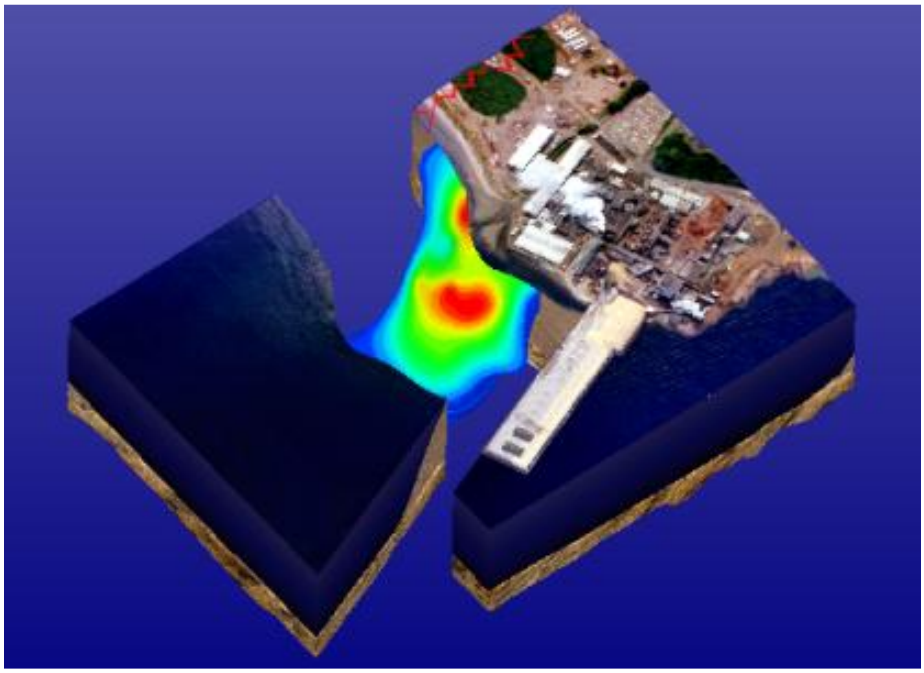
Checking the result accuracy

To determine if the quality of the result is good enough for 3D printing, we need to take a closer look to the model. Zoom in to the corners and details of the model to determine if the surface quality is sufficient. Below, a zoomed in image of the model shows some jagged edges at the originally sharp edges of the model. Also, connection between originally separate shells will show some jaggedness due to the triangle generation based on the voxel structure.

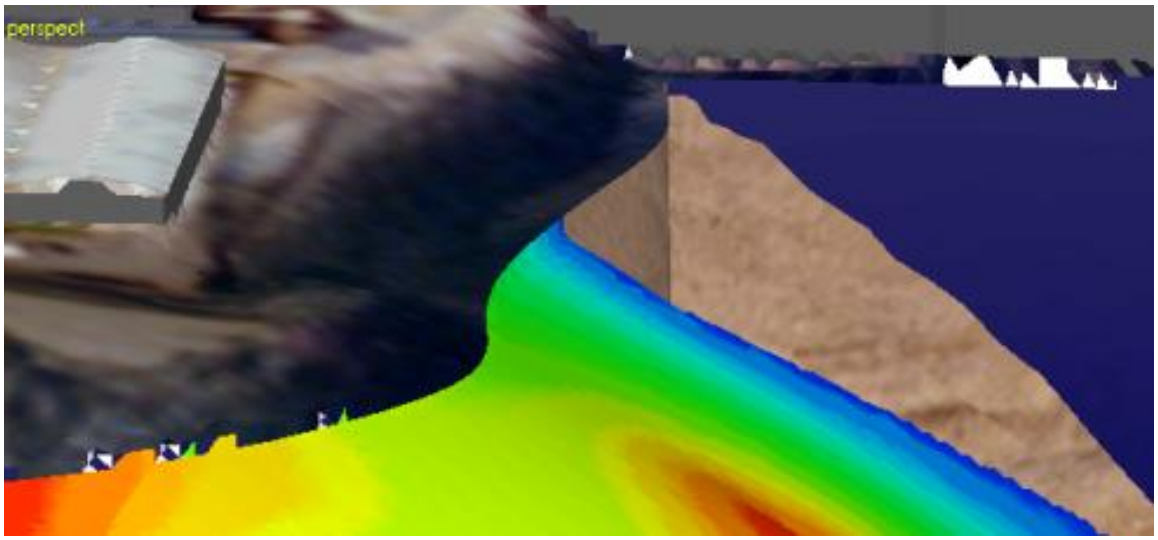
In the image below a measurement is taken from the jagged edge in the model. With this specific model, the dent height is about 0.2 mm. This is close to the layer thickness limit of current 3D Color Printing systems and should not cause any visual effect in the 3D printed part. If necessary, the command can be rerun with smaller Accuracy value (i.e. more voxels and more accurate result). Of course, this will increase the memory requirement by the power of 3, so enough RAM should be reserved for the Solidify command.



Finally, we should check the quality of the color on the resulting model by changing into 3D color printer mode. Below we have the result model shown from the distance in color:

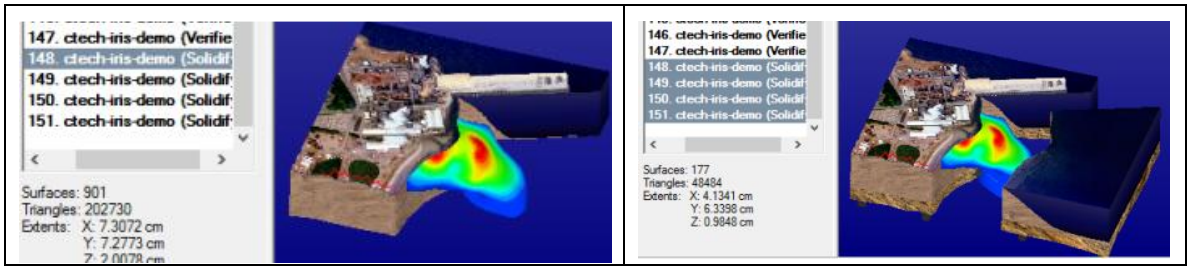


To determine if the quality is sufficient, we need to zoom in to the model again as seen in the image below. Generally the quality is high. There are only a couple of places where the triangles failed to get the correct color. Again, the area of these artifacts is very small and they may not be visible in the final part. On the other hand, these errors can be easily repaired with Paint and Texture tab commands.



Outputting for 3D Color Printing

There are several options to output the repaired 3D model for 3D Color Printing, especially three different formats: VRML, ZPR and PLY. Each resulting shell can be outputted either in a separate file or all in one file. If one shell at a time is required, select one shell and give File Save As command (left hand image). If all need to be outputted in the same file, select all shells with multi-selection and give the File Save As command (right hand image).

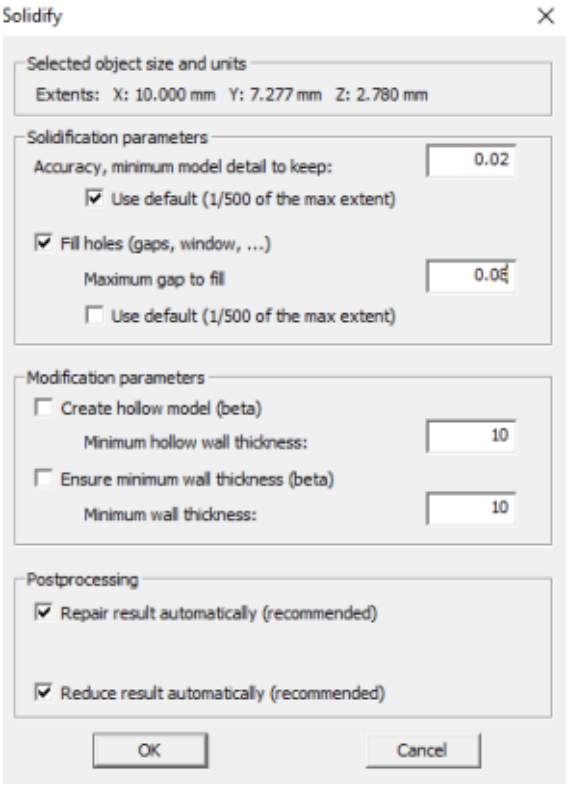


If the number of triangles in the resulting model is too high, the *Modify Faceted* > *Reduce* and *Reduce Color Scan* commands can be used to decimate the number of triangles before saving the results.

Using Fill holes to connect shells

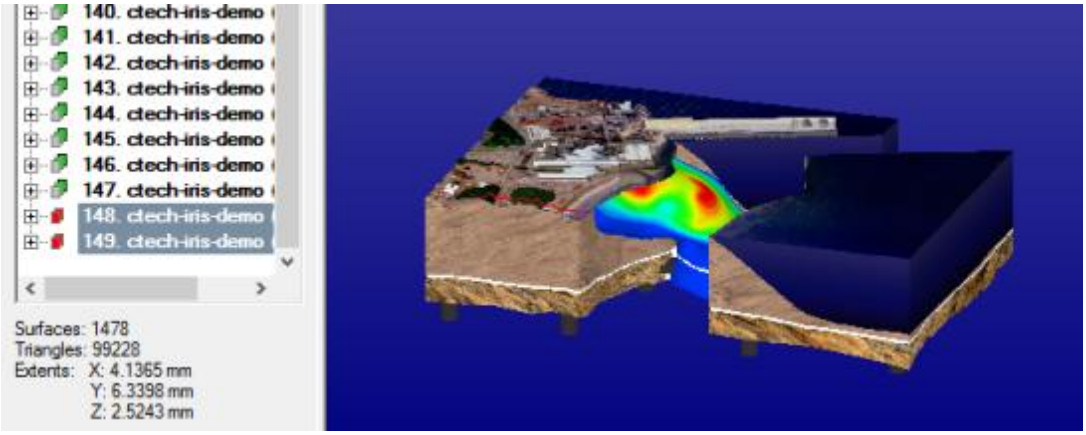
If less than four pieces is required, the Fill holes option can be used to connect resulting shells together. For example, the thin separation between the upper and lower shells may need to be closed.

Running the model through Solidify command with the following *Fill holes* values (*Minimum gap to fill* = 0.8), only two shells are generated:



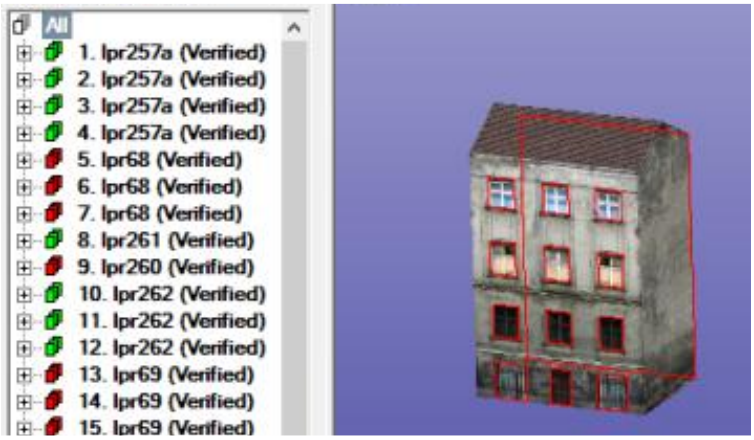
The Minimum gap value can be determined by measuring the distance between the shells with *Dimensions* > *Distance* command.

When the command is ready, extra structure is added between the shells. The new triangles will have the current Foreground color (default is white):

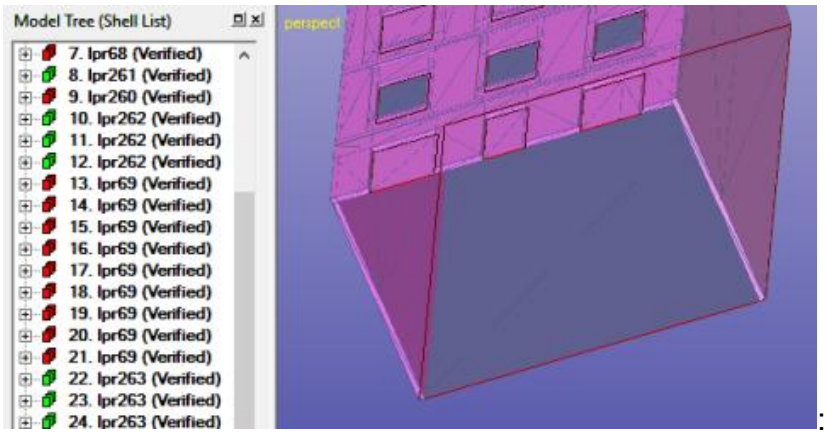


Example 2 - Adding closing surface for Solidify

The second example will be a simple house model with several errors (overlapping surfaces, missing surfaces, duplicate surfaces, solid and non-solid geometry …). The automatic Repair Shells command cannot fix this model properly, so we will try the Solidify command on it.



There is also another problem with the model; it is missing the bottom surface, as seen in the image below (in Repair color mode)



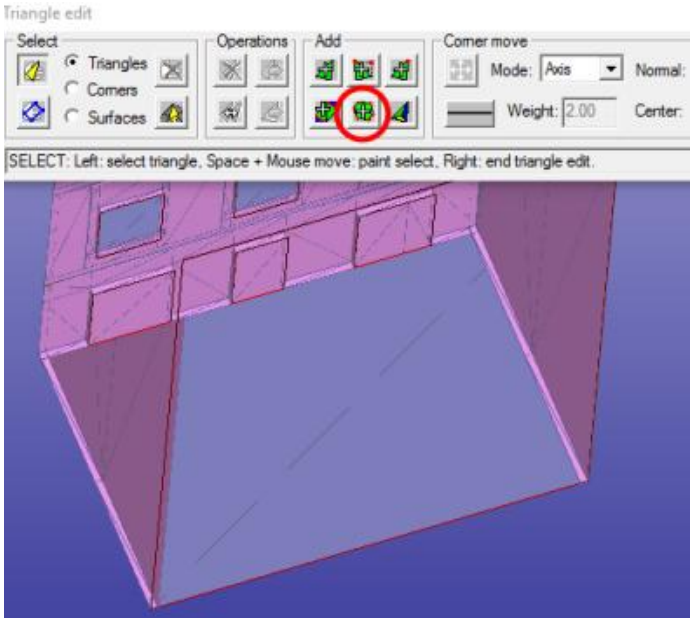
:

The hole in the bottom is too large to be filled with Fill Holes function. It would multiply the running time and also may cause extra features to appear on the model.

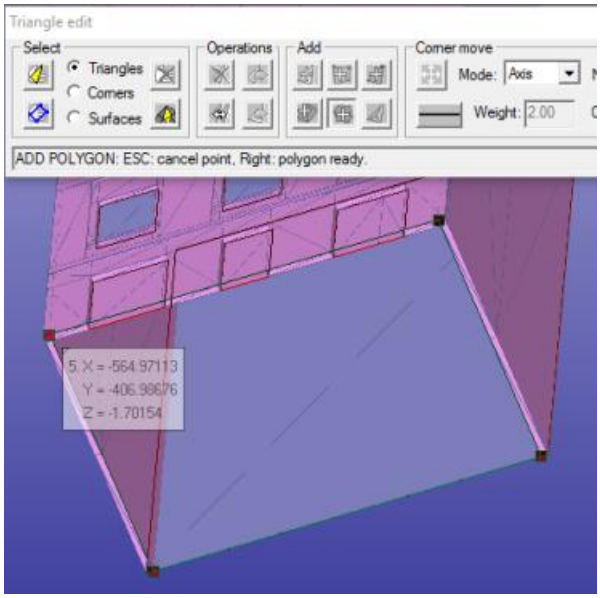
This time we will use the Fix Model > Edit Triangles command to add a closing surface for this hole before running the Solidify command. Let's do the following:

- select ***the Model Tree*** root
- take a ***view*** from the bottom direction to the model
- ***start Edit Triangles*** command

You will see the Edit Triangles tab to appear on the screen:

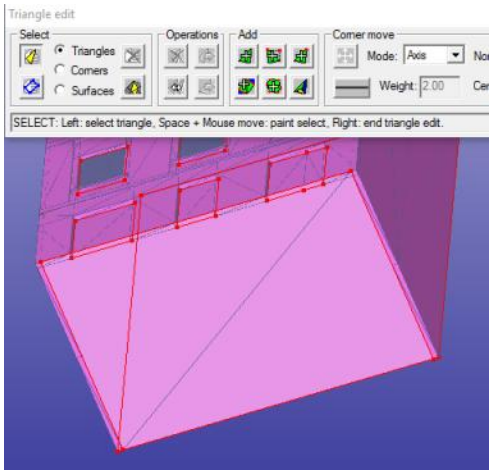


- ***select Add polygon area*** command (in red circle above)
- ***select each bottom corner of the house with the LMB*** (left mouse button) to define a square polygon area



- when ready ***press RMB*** (right mouse button) to accept the new triangles
- you will see a new shell ***Added triangles*** in the end of the Model Tree as well as the closing triangles on the screen

- ***Close the Edit Triangles*** command

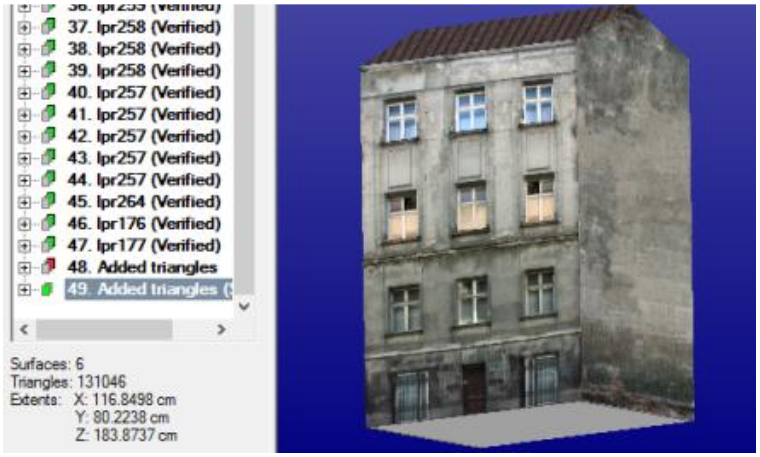


We have now closed the house model bottom for the Solidify command. Let's run the Solidify command now:

- ***select the Model Tree root***
- ***start the Modify Faceted > Solidify*** command
- ***press OK*** to accept the default parameters

The run time is for the solidification is several minutes (depending on the CPU speed) and the memory requirement is about 4 GB.

The end result is a correct solid model for 3D Color Printing.



Other commands, like *Fix Model* > *Fill gaps* or *Join gaps* can be used to close holes in the model before running the Solidify command. This may help to avoid using *Fill holes* with large *Maximum gaps to fill* value as this may cause extra structures to be generated to the result (in narrow passages or dents).

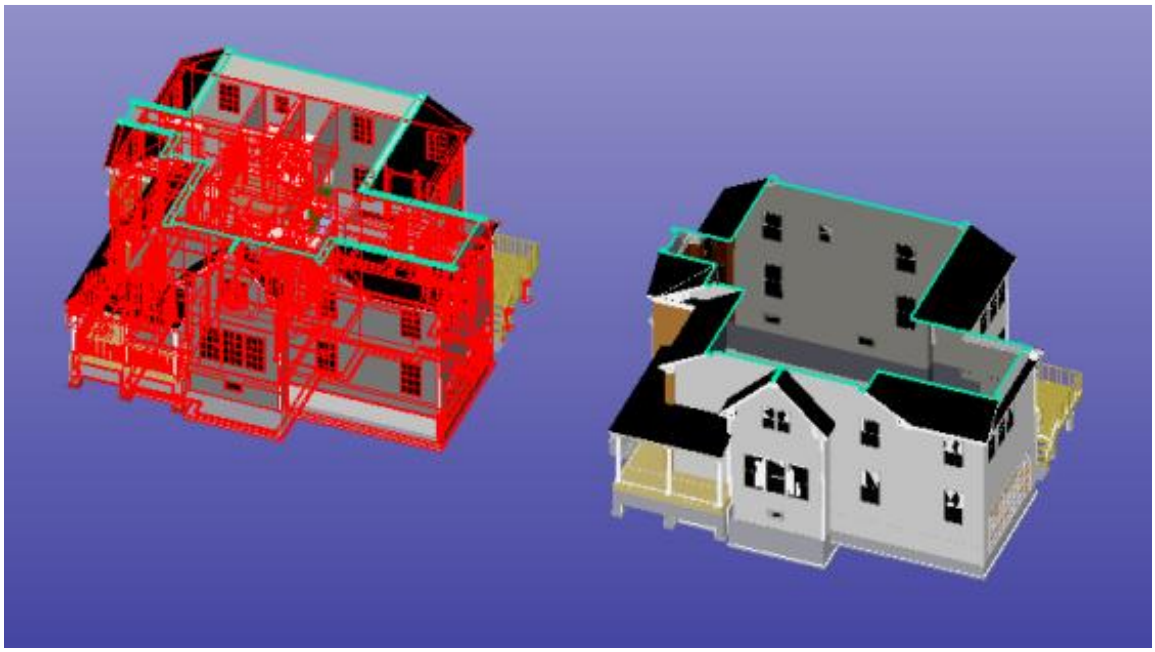
Example 3 - Some development tasks for Solidify

There are some issues which still need to be addressed before the command can be officially released. Some of them are described in the next chapters.

Very complex models with overlapping surfaces

Some algorithm refinement is required for very complex models with gaps of different size and closely overlapping surfaces. Using Fill holes may cause extra structures and color selection in some areas may be difficult.

Anyhow, the command can be used to generate outer shell from various, otherwise totally un-fixable models, as seen below (original left, solid right).



The run time for the model above was 55 minutes (with a mediocre CPU) and memory requirement about 10 GB*.*

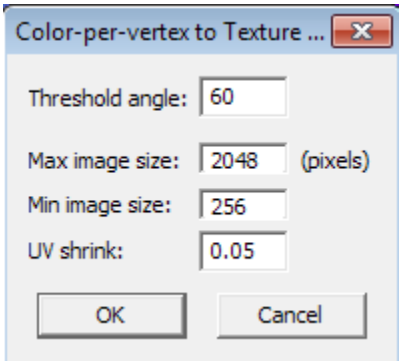
Example 4 – Conversion color-per-vertex to texture

The Color and Texture menu now contains command *Color-per-vertex to Texture.* It converts a model where one or more surfaces have colors defined in triangle vertices to a model where the color information is saved into an image file and texture coordinates are defined for the vertices. Surfaces which have neither color-per-vertex nor texture are converted to solid white (but as a texture image).

The command processes only the model parts selected. This allows the user to convert, e.g., only a specific surface or shell. Non-colored surfaces can also be converted to textured, resulting in a completely white texture ready for editing.

To control the conversion, you can set three parameters:

- Threshold angle: triangles which are at most at this angle compared to their neighbors and the reference normal (the best coordinate axis by default) are included in the same flattening. The default is 60 degrees;
- Maximum and minimum resulting texture image size (square). A suitable size is automatically calculated based on surface detail size. The max and min are used if the value calculated is outside these values. The limits for maximum size are 1024 to 16384 pixels (default 2048), for the minimum size 64 to 1024 pixels (default 256);
- UV shrink parameter: with this parameter the user can control how much the uv values of triangles corners are moved towards the inside the triangle. The value is a relative to the triangle size, not an absolute value. This parameter will probably be hidden from the dialog in future versions.



In the left picture below, a surface of almost a million triangles is displayed as a color-per-vertex model, in the right picture the result after the conversion is shown (parameters used 60 degrees threshold, max 8192*8192 pixels image):

The model consists of one surface only, therefore the surface areas are flattened to a single PNG texture image (8192*8192 pixels):

Known issues with the current version:

- Very small triangles (which would cover only 1-2 pixels) are currently left out of the conversion. They will later be combined to a neighbor sub-image;
- The sub-image stacking algorithm has been changed. The current algorithm needs some tuning to get a tighter combined image;
- Progress bar texts need simplification. Calculation speed is now significantly faster than before so at some phases there's little to see;
- Option to include or exclude very small triangles from the conversion process. This is useful for minimizing the amount of resulting texture images.

.

Example 5 – Texture painting

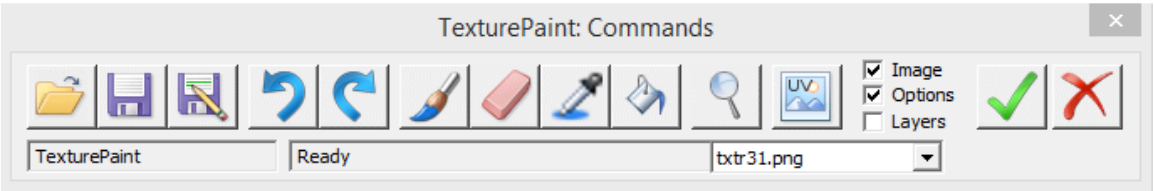
Another new command in the Color and Texture menu is *Texture painting*. This is a totally new feature in 3Data Expert and will enable the user to paint directly on a 3D surface or on a 2D UV-map created from a flattening of the 3D surface. In addition, existing texture maps can be painted on. If an existing texture is modified, a copy of it is created in order not to mess with the original image (which may be used in other models or on other surfaces of the current model).

Texture painting user interface - windows

When texture painting is started, all surfaces within the selection are scanned for texture and color information. Surfaces with no texture (either color-per-vertex models or non-colored) are converted to textured surfaces in the same flattening procedure as used in the *Color-per-vertex to texture* command.

The user interface consists of three windows:

Commands window:



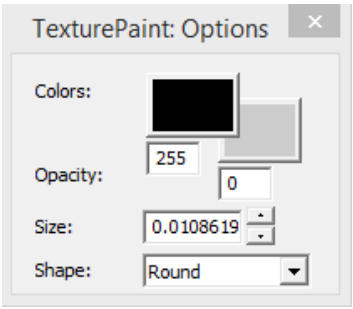
In the current test version, the following commands are available:

- Save the changes and return to 3Data Expert main state (OK)
- Discard the changes and return to 3Data Expert main state (Cancel)
- Brush (the default drawing tool)
- Undo and Redo
- txtr31.png Texture image selection drop-down list. Use this to select which texture image is displayed in the 2D image window. If a new texture is created during editing of an existing texture, the new one is added into this list. You can also click on a surface in the 3D view to choose and display its texture map in the 2D image window.

Options window

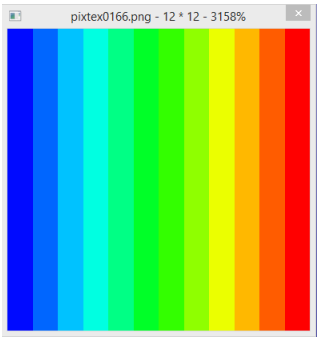
The parameters available for the current operation are shown in and can be changed from the options window.

Currently only the brush parameters are available:



- Brush color (color picker). Now the correct surface color can also be picked from the picker. After the new feature, these alternatives are available:
    - LMB: pick color from standard color dialog
    - RMB: pick color from image (as seen, shading included)
    - Ctrl+RMB: pick color from model. Picking a color from a texture is not yet available, but in case of a texture converted from color-per-vertex the underlying vertex colors are used (interpolated if necessary).
- Brush background color (for patterned brushes)
- Opacity for brush and background color (255: fully opaque, 0: fully transparent). This is not used for antialiasing of edges.
- Brush size. The default size is calculated in 3D coordinates, based on the size and complexity of the model. The up/down buttons double/halve the brush size.
- Brush shape. In this version only *round* is available.

2D image window



The 2D image window shows the selected texture image. To change this image use the drop-down list in the commands window. The window can be resized. The title of the window shows the image name, its size (pixels) and the zoom level.

When drawing into the 3D model, you can select the visible texture image by clicking a textured surface with the left mouse button, or from the dropdown list.

Texture painting user interface - operation

1. Start the texture editing. You can choose the surfaces affected with the normal selection options: single surface, hierarchy (shell, whole model) or multiselection.
2. Choose brush options: color, opacity, size.
3. Start drawing by choosing the desired brush starting point in the 3D window and pressing the **Space** key once. Now the brush path follows your cursor movements. End the brush stroke with another **Space** key press.
4. Repeat adding brush strokes. Do not rotate, zoom or pan the view here! To change the view, first map current strokes to the model with RMB.

5. Update brush strokes into the textures with RMB. Brush strokes crossing surface or texture boundaries are automatically detected and the texture images updated accordingly. Now you can change the 3D geometry view as usual, in order to control drawing accuracy and projection.

   Between the strokes you can also undo and redo one stroke at a time. After the set of strokes has been mapped to the texture, they are considered as one undo entity.

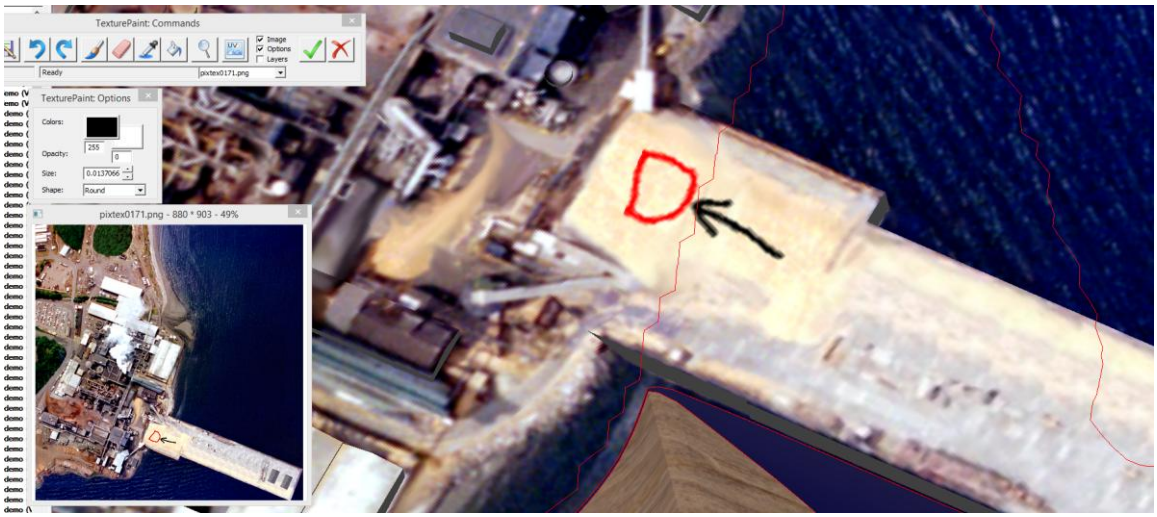
   To get the best result, try to look at the surface as perpendicularly as possible. The more from the side a surface is looked at, the bigger the possibility that non-filled pixels are created into the 2D texture. (Better reliability is under development.)

   Also, with a big model, do not zoom in too close. In the current version this might unnecessarily decrease the performance.

6. Unlimited undo and redo are available. Please not that undo creates temporary files in the Windows user temporary folder. They are normally cleaned after texture painting is ended, but they may remain there if the program crashes during texture painting.
7. To finish texture painting, click either the Cancel button to discard any changes or the OK button to save the changes.

Example

In this example we have edited part of the landscape model discussed in the solidification section. This model is rather complicated, consisting of 147 shells and 3554 surfaces.



Markups mapped onto the texture image.

A close-up of the model, showing different brush transparency effects and the drawing on both the 3D surface and the 2D texture image window.

Test version: bugs

These bugs and missing features will be implemented at highest priority for the next version.

- The brush stroke endpoints may sometimes contain incorrectly drawn polygons which also affect the final texture.
- The resulting strokes as seen in the texture image may look faded or dotted if the target surface normal deviates a lot from the current viewing direction. This is partly a bug but, in general, an as orthogonal view as possible is preferred.
- Speed still needs be improved. Significant improvement has already been achieved but more is needed.

Functionality to be added

High priority: Lasso (“magic wand”, “smart edge detection”): allow the user to select an area for further image operations. The lasso area can also be modified with different operations (“add to lasso”, “remove from lasso”). More brushes and brush parameters, including alpha channel (antialiasing) along edges Automatic resize of a texture is needed when the drawn detail doesn’t fit into the original resolution Filling areas defined with the lasso operations
Medium priority: Drawing into the 2D window Eraser Enhanced color picking for exact reproduction of surface colors Better tracking where painting operations in 3D view affect the 2D view (and vice versa)

Allow 3D view changing between all brush strokes Color picking: allow picking color from texture image (note that in texture painting this can already be done, reliably with the original RMB method, from the 2D image window) Visual feedback which texture image would be affected at the cursor position, as well as choosing the right texture image into the 2D window with a key press (e.g.) Image zoom. The 2D image window can already be resized but an exact pixel zoom is needed for certain operations
Low priority: Layers, for display of flattened geometry (high priority) and image composition Filling areas with a pattern UV editor Option to save the brush strokes as vectors for later editing and property changes