



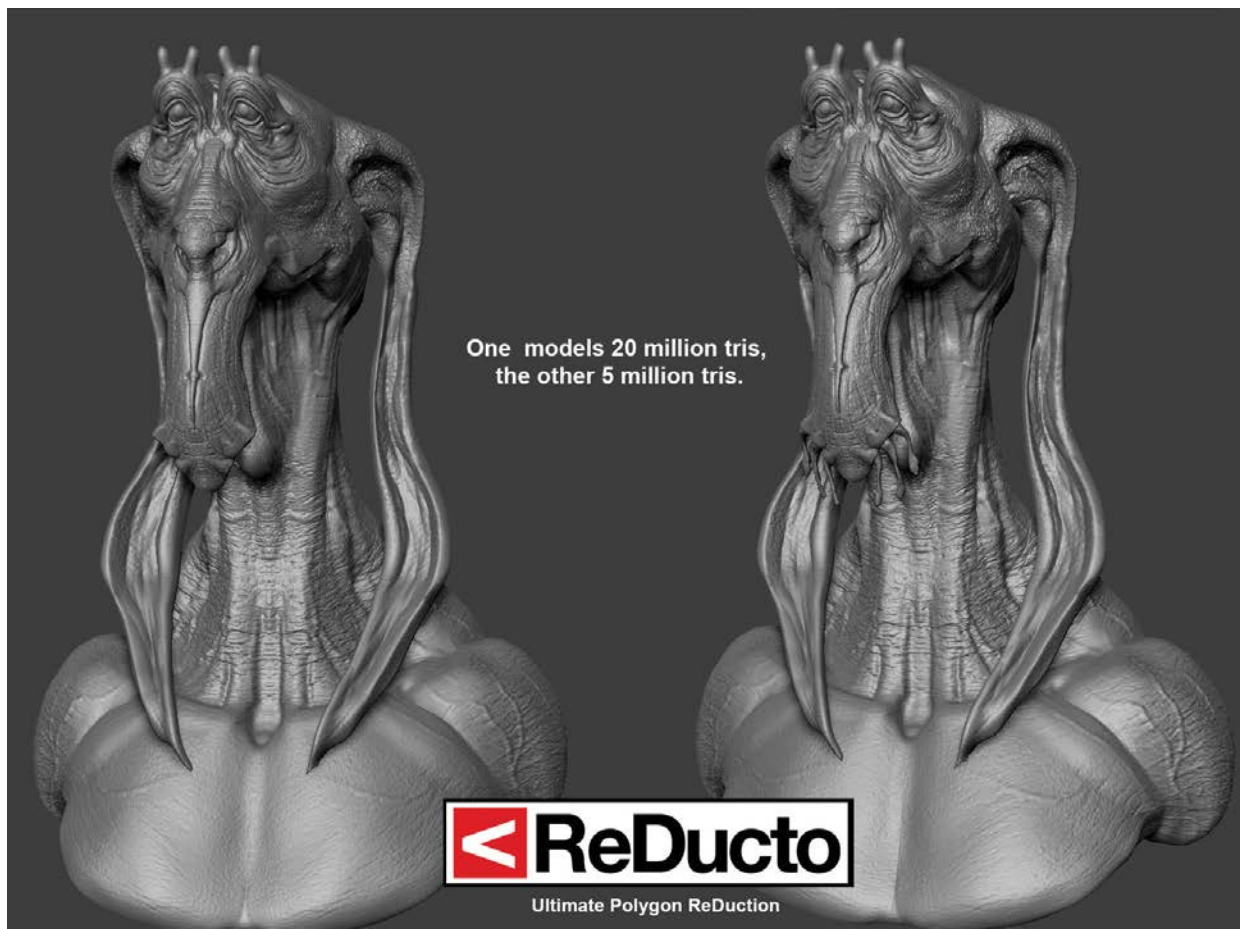
Software



Instructions

CAUTION: ReDucto can be detected by some antivirus software as a virus...it is not. It's a false positive whose solution has remained elusive. All files that make up MudWalker X and ReDucto were compiled on a brand new clean machine unconnected to the internet or any network.

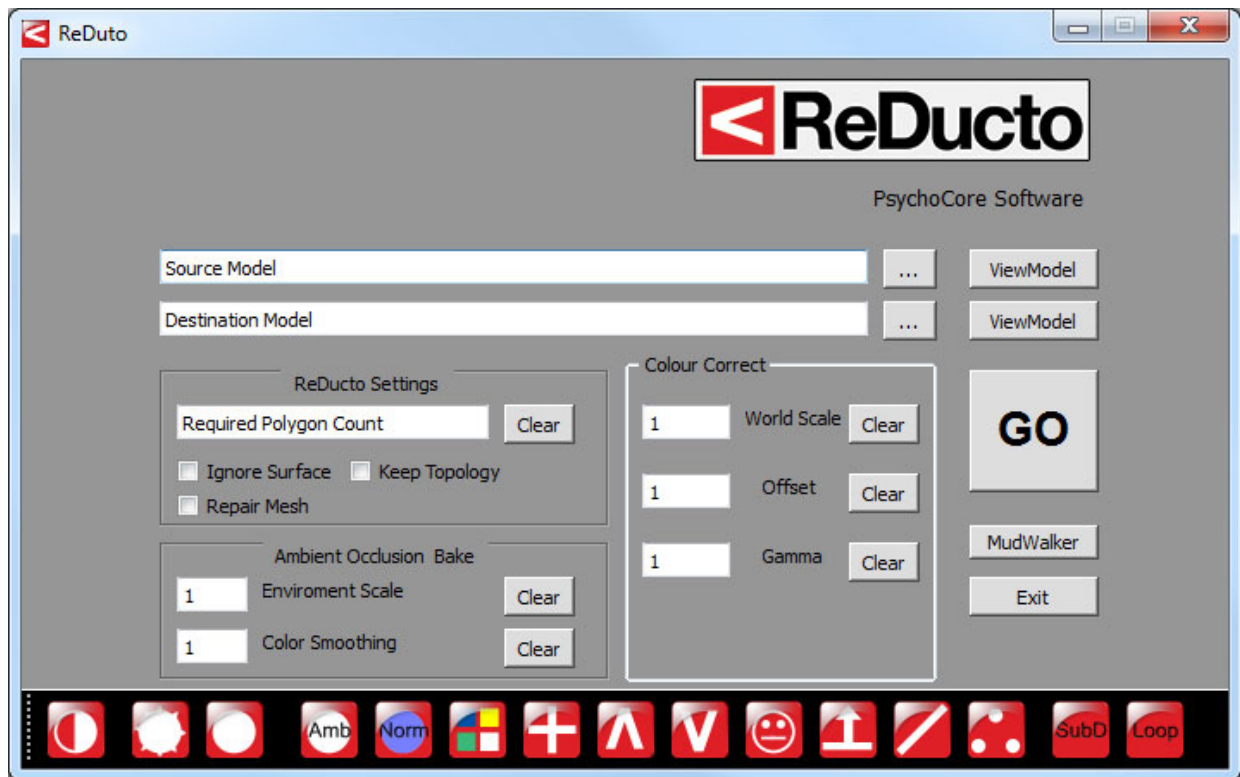
WHAT IS REDUCTO?.....	4
THE TOOLBAR.....	7
TRI STRIPPING	8
TRI UN-STRIPPING	8
OPTIMIZE EDGE CONNECTIONS.....	8
OPTIMIZE FACES	9
OPTIMIZE VERTEX NUMBERING	9
SURFACE SMOOTHING.....	9
FILL HOLES.....	10
MERGE MESHES	11
Bake Ambient Occlusion / Normal's to vertex Color	11
SUBDIVISION BUTTONS	12
THE VIEWER.....	13
<i>The Navigation keys</i>	13
‘NoQuarter’	14
SUPPORTED FILE FORMATS	14
REDUCTO CORE: POLYGON REDUCTION	14
USING REDUCTO’S POLYGON REDUCTION	14
Normal’s.....	15
Keep Topology	15
Repair Mesh.....	15
THE RULES OF REDUCTO	15
3D PRINT WORKFLOW	16
UV MAPPED WORKFLOW	16
KNOWN BUGS!!(Or how to break it)	16



Well... can you tell which is which?

(This file is available as a sample on request)

WHAT IS REDUCTO?

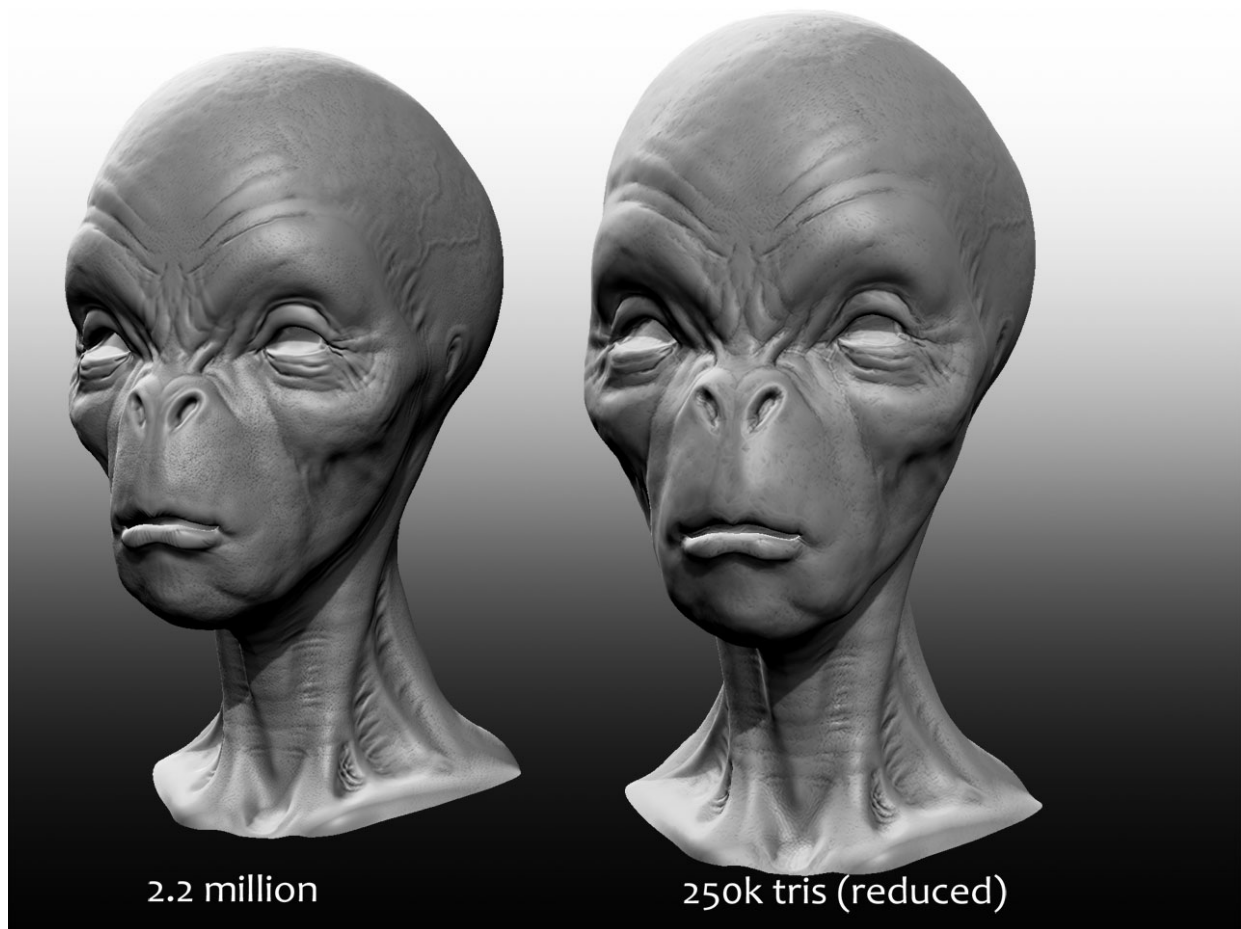


What is 'ReDucto? It is the 1st new polygon reduction algorithm in many years (99% of all others are based on the same quadratic reduction algorithm). ReDucto was programmed and devised by Wayne Robson under a total black out over the space of more than 3 years in his spare time. As a digital sculptor himself, he knows far better than someone who is not a sculptor what polygon reduction needs to do for other digital sculptors.

So it is hoped that it provides you with a toolkit that can solve many problems without solutions, and solve ones with existing solutions in a far better way. ReDucto is a standalone app that can be tightly integrated with almost any 3d application or pipeline that requires its tools. ReDucto only ships as part of the MudWalker X plug-in suite for Mudbox.

I created this from scratch using my own unique algorithm, so there is nothing like it anywhere. originally it was penciled for a commercial release but after a backer let me down...well I never released it. Now I'm making a un-optimized version of ReDucto available to everyone for free as it seemed such as shame only a handful of beta testers using it. It has been used in production now by beta testers for nearly 3 years without problem which should give you some peace of mind. The fully optimized edition will not be released as that is my own personal tool and now contains things far above what ships in this version. As a freelance artist that give me an edge no one else has...sorry.

Also worth noting is that ReDucto's algorithm is unique and was devised by Wayne Robson and features in no other application. Also you may notice that it's feature set goes FAR beyond polygon reduction, that's only part of the story.....

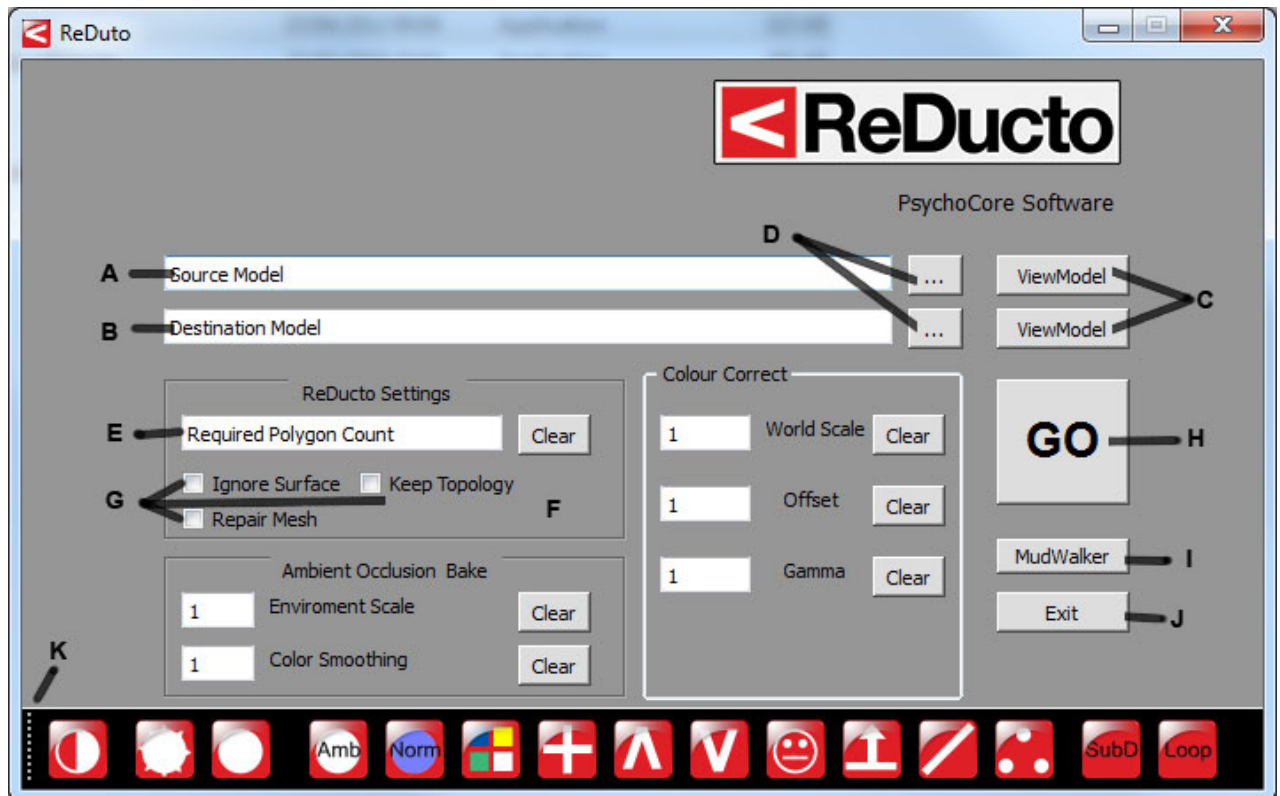


The very 1st public test model from April 2009

It can not only optimize a model so that it loads faster, it can solve a number of mesh problems that could cause issues further down a pipeline. On top of this it features an adaptive hole filling tool that will make a mesh watertight in a single click using an adaptive polygon flow. *(That to say it will add only the amount of polygons needed and using its internal AI, it will try to respect the surface curvature of any mesh you want holes filling in for.)* There dedicated triangular subdivision tools vertex color baking tools and more.

As ReDucto is part of the MudWalker X suite *(that is also free)* for Mudbox users, it also provides for the first time Mudbox with its own polygon reduction.

Bundled with MudWalker X and ReDucto are a couple of very useful tools for any 3d artist . A drag and drop model viewer and a nice mini app that you can run to get rid of all Mudbox settings files that quashes many of the most common bugs that occur to Mudbox users. *(Mudbox then creates a new set on start up)* While the viewer speaks for itself what it is for, 'NoQuarter' is an algorithm that is totally separate from the main ReDucto polygon reduction algorithm. It behaves in a unique way that either people working with insanely heavy polygon count meshes or hard surface modelers & those working with production parts will find invaluable. Both of these two bundled applications are simple to use and require no special knowledge on the part of the user. They are just about idiot proof.

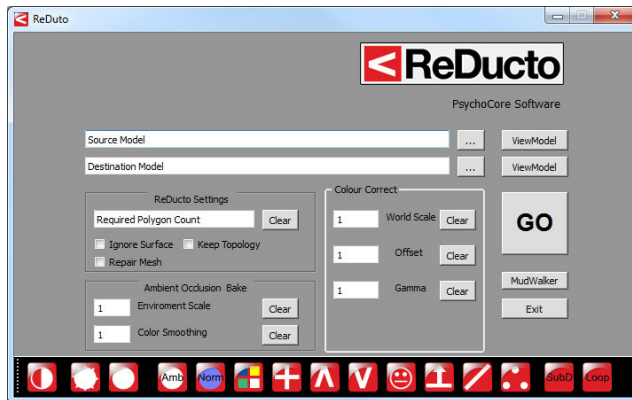


- a Destination of Source model
- b Destination of Output model
- c View Model using ReDucto Viewer
- d Browse for model
- e Required Polygon count
- f Clear Required polygon count
- g Options
- h Reduce the polygon count using settings in UI
- i Import from MudWalker Supported Apps
- j Exit ReDucto
- k Drag to undock toolbar

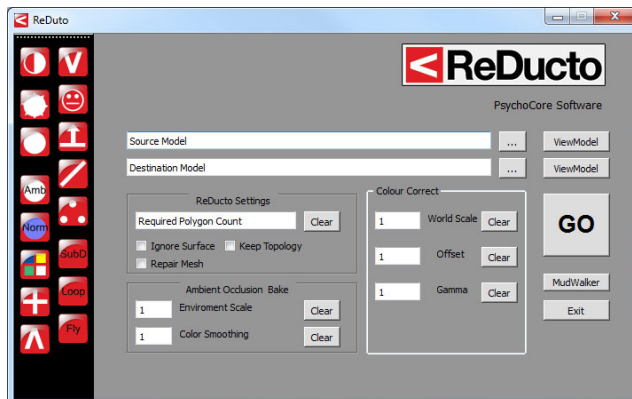


- | | |
|--|--|
| 1 Adaptive intelligent hole filling | 9 Industrial strength tri un-stripping |
| 2 add noise to mesh (<i>removed feature</i>) | 10 Fix mesh problems |
| 3 smooth mesh noise | 11 Optimize Faces |
| 4 Bake Ambient Occlusion to vertex color | 11 Optimize Edges |
| 5 Bake tangent normal to vertex color | 13 Optimize Vertexes |
| 6 Color correct vertex color | 14/15 Subdivide source model using SubD Loop type 1 & type 2 |
| 7 merge two meshes into single mesh | 14/15 Subdivide source model using Fly Subdivision |

THE TOOLBAR



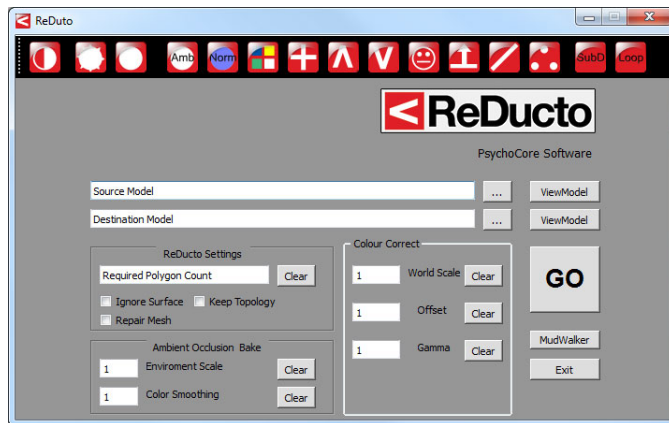
Default Bottom Docking



Toolbar Docked at left-hand side (other buttons available through arrow)



Toolbar Undocked at and placed above



Toolbar placed at top of UI

ReDucto's toolbar set by default at the bottom of the UI can be docked in any location on the interface and even detached all together and left floating where you feel you need it. This toolbar contains the expanded toolset features of ReDucto ranging from model optimization to vertex color baking and beyond.

TRI STRIPPING

ReDucto contains an industrial strength tri stripping feature that has been tested internally on meshes of a truly huge polygon count. Chances are no one will ever need to tri strip a model of that size, but you can be sure it can handle the data you put into it.

TRI UN-STRIPPING

This is the same as the above but in reverse. This feature is included only for those who really must un-strip a model and again it has been tested on very large polygon count meshes.

OPTIMIZE EDGE CONNECTIONS

To help your model load faster, whether it is in a game engine or some other target format, ReDucto contains the Edge connection optimization feature. This will optimize the way all the edges in your model are connected to ensure your model loads in a logical way that speeds up loading times.

OPTIMIZE FACES

This is the 2nd part of the mesh optimization group of tools in the toolbar. This time is concentrates on all the faces in our model and works its way around the mesh finding and fixing where it can any problems that would slow down loading times. On heavy models this can be a massive time and resource saver.

OPTIMIZE VERTEX NUMBERING

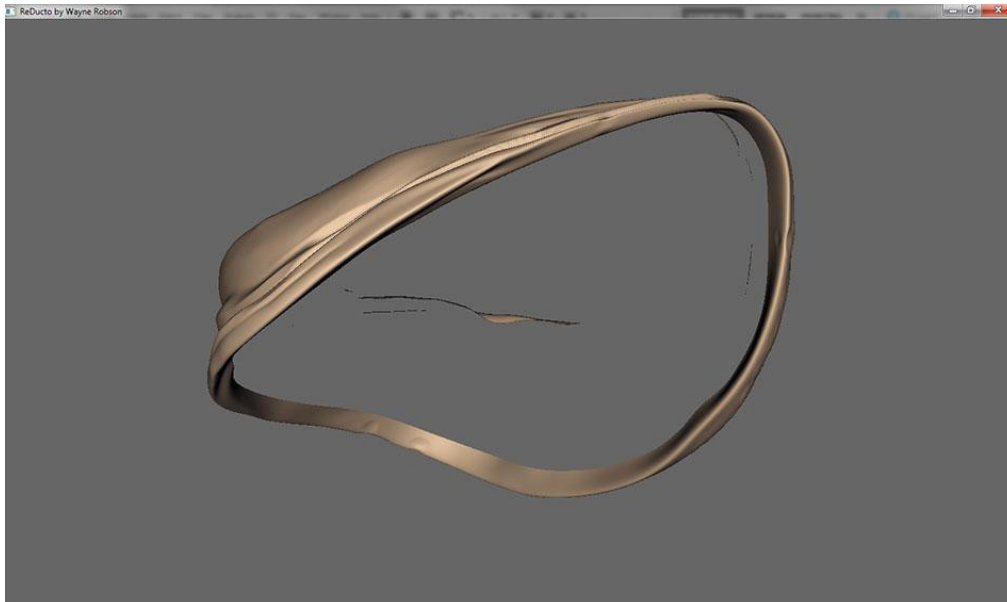
Many models output by 3d applications do not have a logical system for numbering the vertexes in a model. So as an example vertex number 23456 may be the end of a characters nose, where 23457 may be on his big toe. This often mean son loading that a lot of jumping around s done in 3d space on loading. This can slow down your models loading time. More importantly should anything go wrong with your model this means that you can save parts of your model in a logical fashion. As ReDucto numbers the vertexes from the top downward you would be able to save far more usable data than in any other way.

To visualize in your head how ReDucto numbers vertexes imagine filling a glass full of milk...only looking at it upside down. On heavy models this vertex numbering can have a massive effect on loading times and on extremely heavy models it can even allow a model to load that would otherwise crash a program.

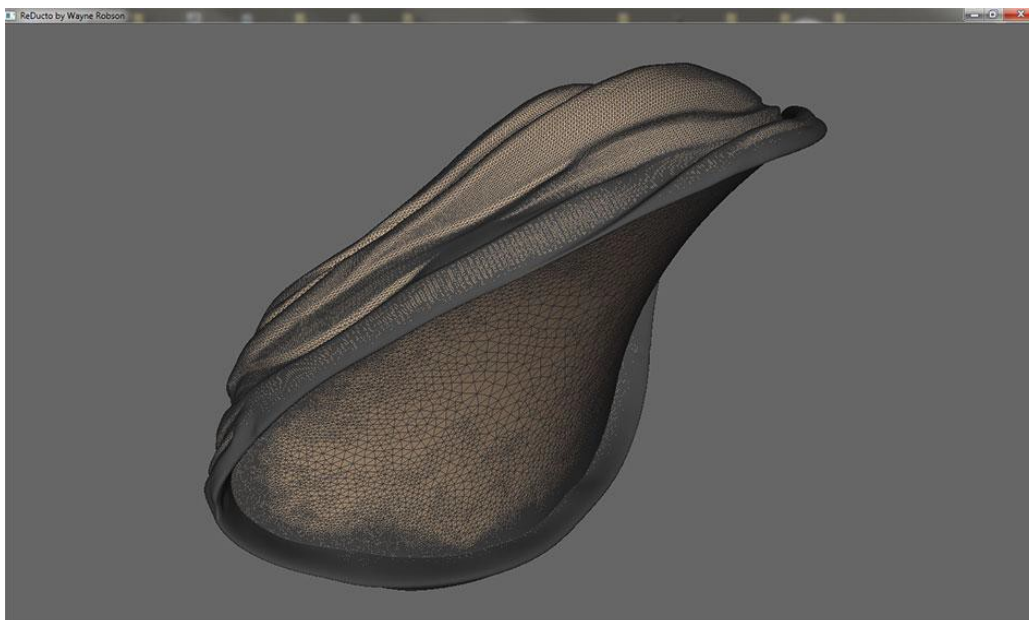
SURFACE SMOOTHING

Surface smoothing should not be confused with subdivision. An example of smoothing would be a mesh taken from scan data that may contain quite a bit of unwanted surface noise. The ReDucto smoothing algorithm allows you to try and reduce this before you reduce its polygon count.

FILL HOLES



Now this is what I call A 'ROCKSTAR' feature as it is one that is both very useful and took a long time to perfect the algorithm for. The hole filling tool will search your mesh and find any holes in and fill them in an adaptive way using only the polygons it needs. This is perfect if you have a model that you wish to have 3d printed but it contains holes (such as a bust using the Mudbox default head base mesh for example).

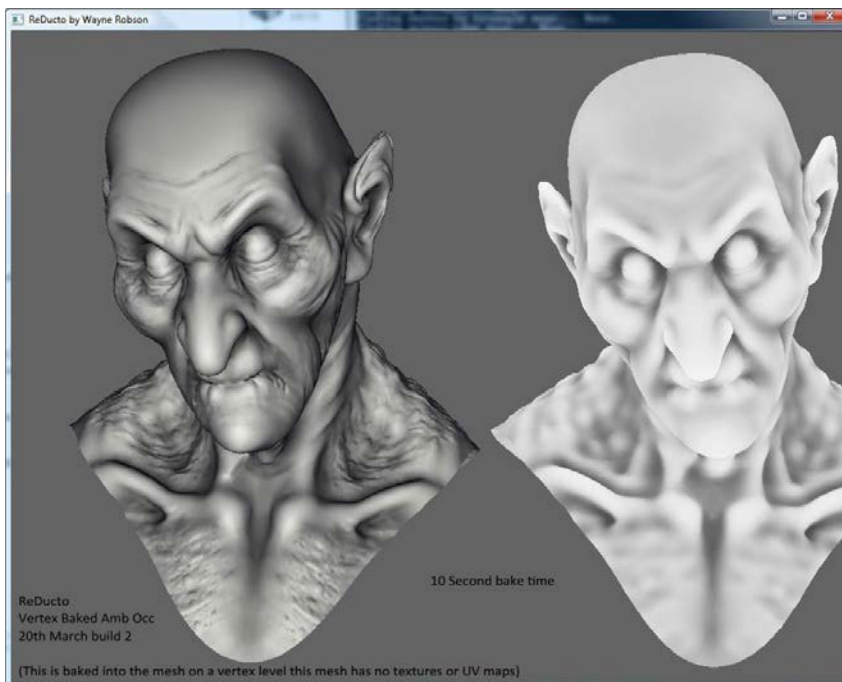


3D Printing machines require 'watertight' models with no holes within them...this tool enables you to create watertight meshes with the minimum of fuss. As it doesn't throw any more polygons than it requires at the resulting hole filling it also helps to keep the resulting polygon count down. When filling a hole it will do its best using its simple internal AI to respect the surface direction and maintain any curved surfaces that it can.

If you come across a problem it is likely due to one thing and one thing only, if you have a point in a section of your model you have exported that has overlapping faces it can give funky results... But simply smooth that area out to get rid of this and the funky problems disappear. This is a modeling problem not a ReDucto problem by the way.

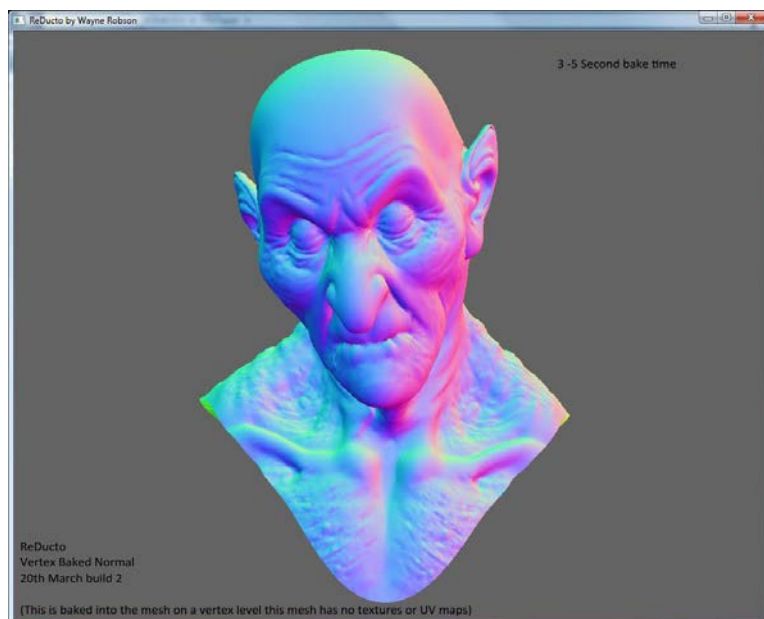
MERGE MESHES

The premise of this tool is a simple one, it takes a mesh in Source slot and a mesh in the destination and merges these into an auto named file saved to the *C:\MudboxTemp* folder and named *MERGED.obj*. As ReDucto works on a single mesh at a time, if your model has many parts and needs to be a single file for printing at the end, this enables you to bring all the optimized parts together into that single file right at the end.



Bake Ambient Occlusion / Normal's to vertex Color

The vertex color baking in ReDucto has a few possible uses. Vertex colors can have a lot of uses in game engines for use as masks etc. When color is baked to a vertex instead of a texture, it means that a texture map doesn't have to be loaded (as the color data is loaded long with the mesh, this means a time and resource saving.)



This feature is similar to the ambient occlusion color baking and is included as a 1st step towards a future feature. I was asked to include it in this build due to request received from a beta tester. This tangent normal color data can be used in a few ways to generate other map types from it, or to pull data for the mesh in an easier way. I've left it in purely in case someone finds a use for this somewhat strange feature in their pipeline.

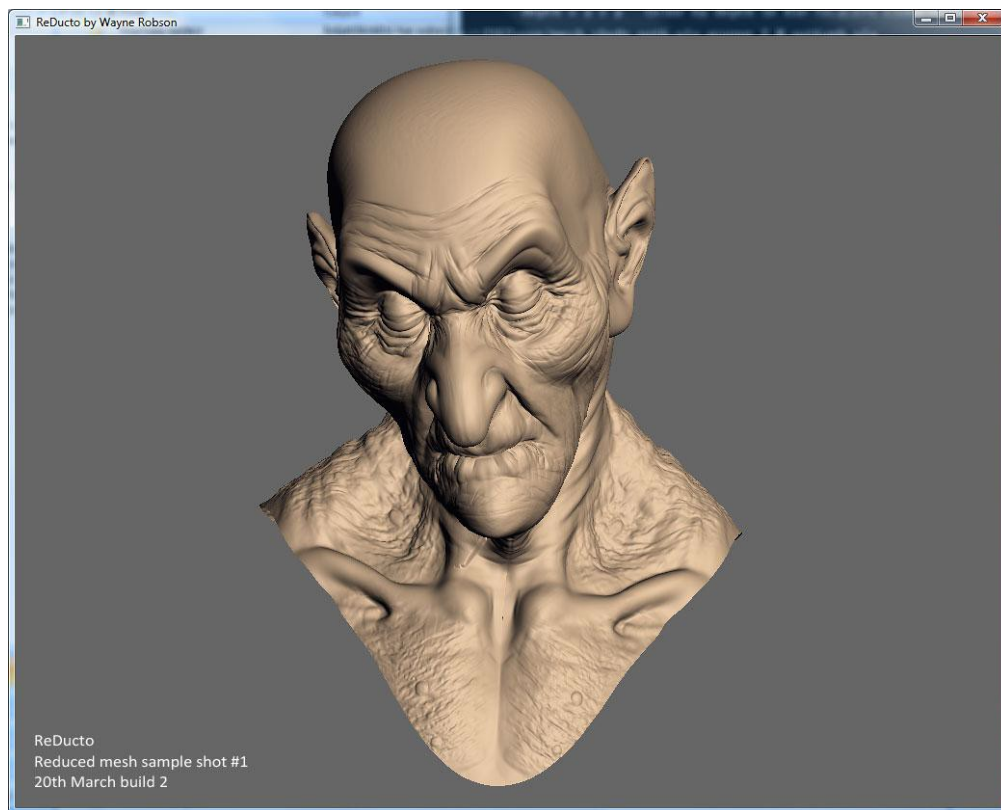
There are actually two independent algorithms in ReDucto to solve mesh problems such as stray vertexes etc. As well as this one in the toolbar there is also an option in the main ReDucto window for use when reducing your models polygon count.

The thought behind this is simple. If one doesn't solve your problem, there is a very good chance the other one will. (Plus both work very well and it seemed a shame to leave one out.)

SUBDIVISION BUTTONS

The 3 types of subdivision supplied in ReDucto are aimed at triangular meshes. There can be times when a reduced mesh is simply to 'jagged' for the use you intend to use it for, these allow you to sub divide our model in one of 3 ways to keep the polygon count as low as it can be, but also smooth the model.

THE VIEWER



The viewer is shared across a number of tools both in ReDucto and MudWalker X and is also available as a mini app outside of these I've included as an extra.

The separate model viewer has also been made available as a separate mini program and can be used to simply drag and drop model files onto it and so it can be used as a very fast way to preview files.

The Navigation keys

Key

Left Mouse Button
Right Mouse Button
Left + Right Mouse Button
Middle Mouse Button
Ctrl + Left Mouse Button
W
L
Spacebar
B
S
D

Result

Rotates the model base on its pivot point
Zooms the model towards and away from the camera.
Pans the camera
Menu
Move Light
Wireframe Mode Toggle
Flat Lit mode Toggle
Re-center the model in the scene
Background Color Toggle
Specular Shader Effect Toggle
Double sided Mode Toggle

I created the ReDucto Viewer to hopefully take the best parts of both the main digital sculpting applications and create a viewer that was both intuitive to be easy to learn and had the grunt needed. The viewer is limited in the polygons it can show depending on your system and graphics card. The viewer will be refined during future betas.

‘NoQuarter’

NoQuarter is not ReDucto. It is a separate entity using a unique polygon reduction algorithm that is useful either for those working with insane polygon counts that no other reduction app can handle, or for manufacturing or hard surface models. It is available directly from the MudWalker X Interface with 9 easy to use preset levels of reduction and creates as near as it can a totally even triangular polygon flow. It uses a voxel grid based collapse system for the reduction so is not as accurate or give as excellent results as ReDucto can.

NoQuarter can handle truly massive polygon counts far in excess for the main ReDucto app. So if you have a mesh that is of a horrific polygon count, think of this as your get out of jail free card. It can also be used as a usable mesh either for baking to a retopologised mesh or before running it through ReDucto itself. ...and yes it is named after the Led Zeppelin song of the same name as in its early alphas it reduced the mesh polygon count by a tiny bit more than a quarter of the mesh size.

SUPPORTED FILE FORMATS

ReDucto supports a number of file formats, but the 3 currently exposed for users in this version are:

Wavefront OBJ

PLY

STL

The last two being specific triangular mesh formats will enable you to plug your model straight into a 3d printer with no bother at all and retain any baked color information you have created in ReDucto. They have the added advantage of not containing much of the ‘junk data’ that can be included in an OBJ meaning smaller file sizes. other file types can be used by exporting from MudWalker X in Mudbox or Supported apps.

REDUCTO CORE: POLYGON REDUCTION

Polygon reduction is ReDucto’s whole original purpose for being created. In this section we will explain how to use it, how to get the best out of it, and even how to break it! (not many software companies how you can break intentionally their product.)

USING REDUCTO’S POLYGON REDUCTION

Using ReDucto ‘out of the box’ after an install is a fairly painless process of putting your high resolution source model into the source slot. You can do this by browsing of the file using the browse button. You then do the same thing to tell ReDucto where you wish you reduce model to go along with its format. (By default ReDucto will

export as a Wavefont OBJ unless you specifically tell it otherwise.)

There are only 3 options exposed in the ReDucto UI to hopefully keep things simple to use for end users. Lets outline what each does and how it affects the final output.

Normal's

If you wish to let ReDucto look at the meshes vertex normal's and check this button. If you do not, it will then the AI to work it out by itself which can often give fantastic results. Unlike a quadratic algorithm where the normal's give the information needed for polygon reduction, ReDucto uses this only a sort of a kick start and way of cross checking results against the original source mesh at various stages. Standard polygon reduction algorithms were intended originally for use on large flat data planes from the oil industry and were not intended to work on a complex sculpted 3d model.

Keep Topology

You can as ReDucto nicely to try and keep the topology already within your model for its reduction. This means that all edges will obey the same polygon flow as your original model, only in a more adaptive way. If you ignore this, then ReDucto uses your mesh as a reference while it reconstructs a brand new mesh from scratch using its artificial intelligence to work out the best way to create your mesh using the polygon count you have specified. You will notice that ReDucto 'learns' by each reduction it does and eventually it is intended that it tries its best match the way you model , or the models you regularly put into it are made.

Repair Mesh

You will find that ReDucto contains not one but two mesh repairing algorithms; both are completely different and based on different algorithms of mine. One is found in the toolbar and it outlined earlier in this help file. The other is part of the ReDucto core that is used what you use ReDucto Polygon reduction. This version lets your mesh be check for problems before reduction takes place. This helps to spot anything that would slow it down or affect the output.

THE RULES OF REDUCTO

1. Ensure that the mesh you input into ReDucto is a single part only, you may recombine them later using the Merge function
2. Remember the bigger the difference between your high resolution and required polycount, then the longer it will take. It is an application not a magic wand.
3. If you intend taking your low res resulting mesh into a game engine, then remember you will need to apply smoothing groups in your 3d application of choice.
4. No writing on the wall
5. Obey all the rules.

3D PRINT WORKFLOW

Export your high resolution mesh from your app of choice. If you have the chance then try PLY or STL format as these are smaller and carry less bloat. ReDucto can handle OBJ files very well though. Find out if you can what the maximum tri count the machine your model is being printed on can handle and what formats. Aim for slightly below this as your maximum for your reduced mesh.

Split our model into parts if it is made of more than one mesh. If you want to you can also split a very heavy polygon count mesh into smaller parts to enable faster workflow should your app support this. Put your part or parts through ReDucto and remember you are dealing with a tri count not a quad count. So if your high res mesh is 4 million polys in Mudbox or Zbrush that will equate to 8 million tri's.

Don't be afraid to do a couple of versions with different settings and poly counts to get the best result you can. Recombine your files (if you have more than one using the merge tool in ReDucto. This combines the models in the source and destination slots into one model file. Don't forget that 3D printing needs a watertight hole free mesh so make sure you have done a clean up on your final model /models using ReDucto's mesh clean up tools and fill any holes using ReDucto's adaptive hole filling tool.

UV MAPPED WORKFLOW

ReDucto currently dumps any UV mapping that your high resolution model may have. The reason for this is a simple one, that depending on if you have 'keep topology' switched on or not it either uses the mesh itself and reduces it, or creates a brand new one from scratch. Added to this poly reduction takes a while and it takes far longer if UV mapping has to be taken into account as this means data has to be dealt with for every vertex and face in a mesh.

So how do we get some UV's on our reduced mesh? Nearly every 3d application worth its salt has a feature to transfer UVs' / transfer maps. So you simply export a medium resolution mesh with UV's that conforms pretty closely to our reduced mesh and transfer the UV's from one to another. In this way you can then use a resulting reduced mesh in a game engine (tests have been done in production of this method using ReDucto for both the Unreal and Unity game engines with great success.)

KNOWN BUGS!!(Or how to break it)

- ReDucto will not work with OBJ's output by Nevercenter's Silo due to the way they output the OBJ format....they can fix it their side as I am not. (The chances that you will ever need to reduce a silo mesh though are minimal anyway.)

- A weird QT bug meaning ReDucto will not work if a file path has a space in it at any point. These can also give pipeline problems in other apps as well that use QT. (Qt is a library for UI creation it also does not support 64 bits as of yet so that is why ReDucto and MudWalker X run from a number of separate exe files.)
- ReDucto only works on a single mesh at a time. If you input a mesh that contains multiple merged parts it can reduce the polygon count in unexpected ways. Multiple part mesh input is not supported due to the algorithm it uses. ALWAYS separate your part before using ReDucto.
- ReDucto is not super fast, it takes its time in the version you have (mine is far faster), firstly because it is free and secondly because this extra time provides better results.
- Some virus scanners can see ReDucto as a virus...it is not, I have no idea why this false positive occurs after 2 years of trying to stop it. If it gives you any peace of mind the compile you have was done on a brand new machine with fresh installs only.