Voxon Media Creation Guide

The Voxon VX1 can display many types of 3D content. *VoxieOS* is Voxon's primary media player, supporting many common 3D file types such as .OBJ, .STL, .PLY, .KV6, KVS, .MOL, .FBX (via Unity), DICOM, .DICOS, .JPG, .PNG (including heightmap), and .KNI. KniView can be used to capture and playback depth camera footage while custom VX applications can be developed in C/C++ or using *Unity* via the *Voxon Unity Plugin*. *Unity* has a vast community and with plugin support, it is often used as a means to view 3D media.

This document will walk through the process of creating and viewing content for *VoxieOS* and *KniView*.

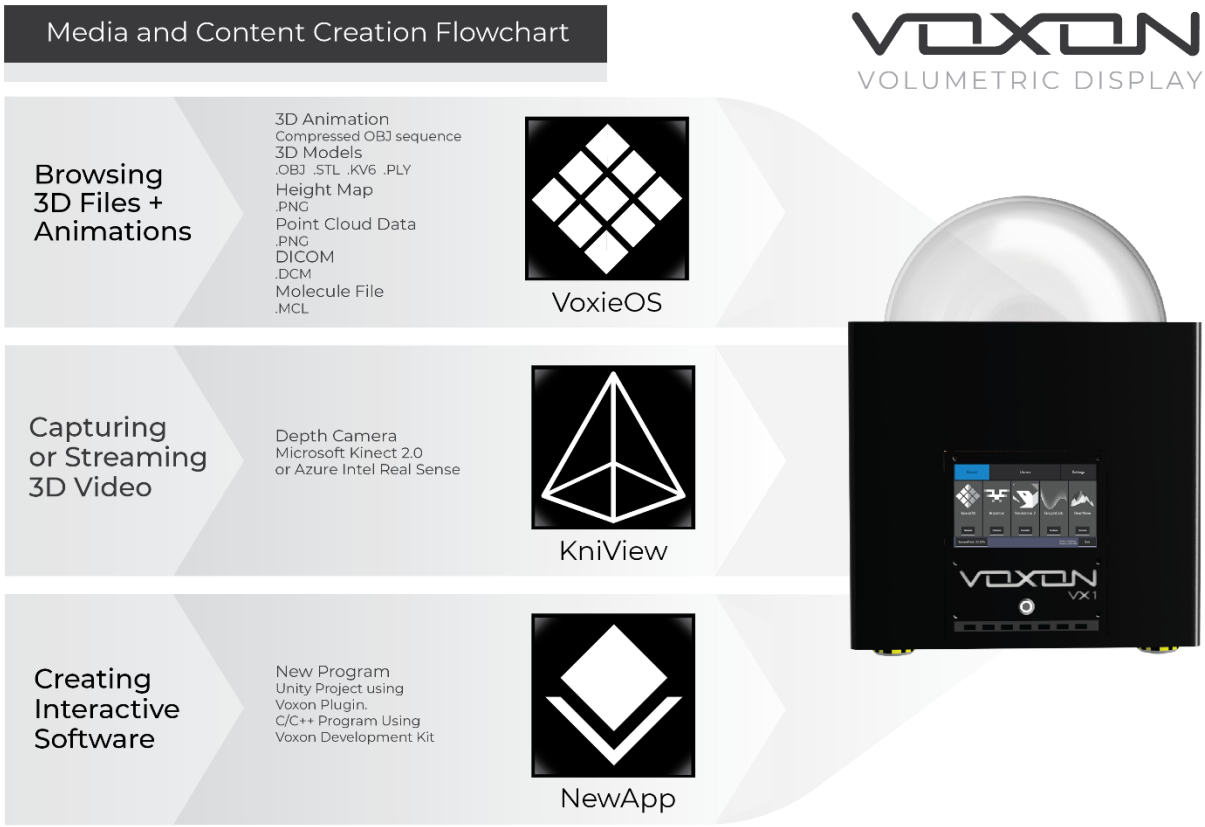

Figure 1 Media and content creation flow chart for Voxon volumetric technology

Navigating and Viewing Content

The *VoxieOS* interface is designed to be like a '3D file browser'. By default, it's starting directory is located at 'C:\Voxon\Media\'. after being launched, *VoxieOS* will show the file content of its starting directory. This can be changed by editing the *voxieOS.ini* file.

We encourage Voxon users to place their media content inside the C:\Voxon\Media\MyMedia\ folder. However, you can obviously manage your media anyway you like as they are just files in folders.

To interact with *VoxieOS*, you can use a keyboard, mouse or the space mouse. Key bindings and commands are displayed on the secondary (touch) screen depending on the type of media being viewed.

Use the file browser to navigate to the media content to view.

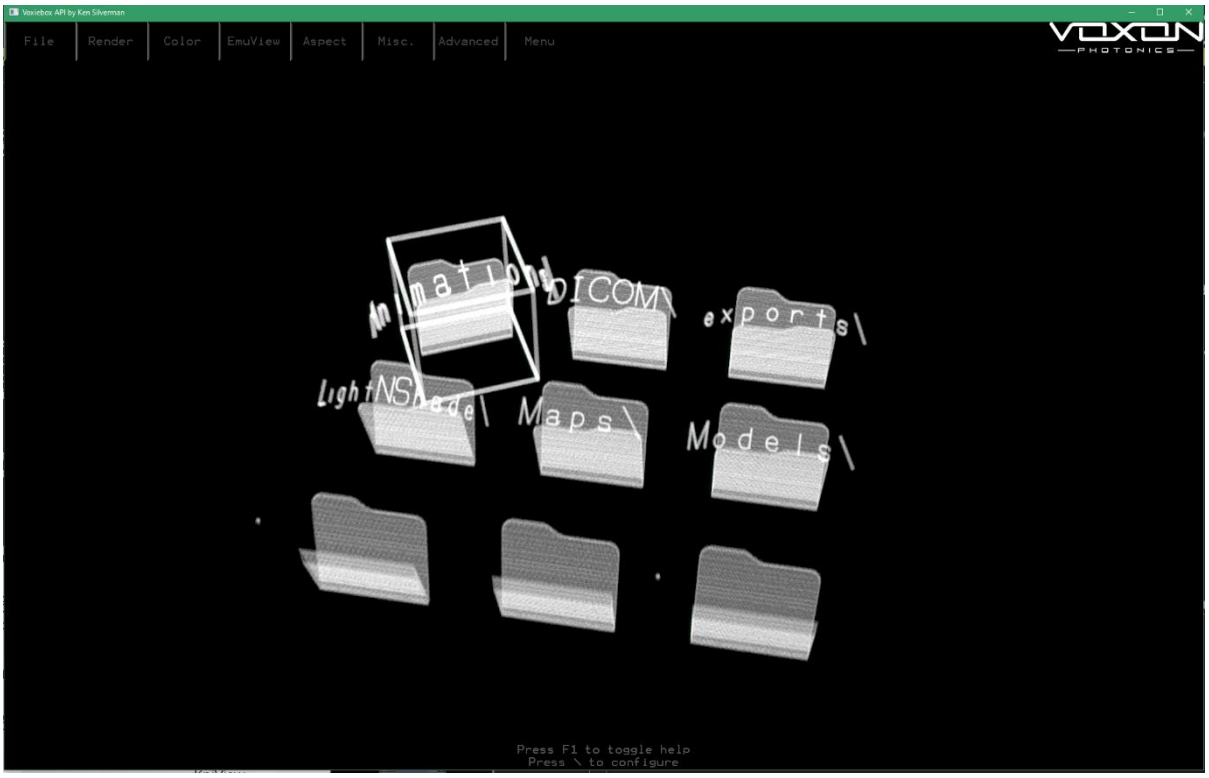

Figure 2 VoxieOS's file browser.

Mounting a USB Drive

If a USB drive is inserted while *VoxieOS* is running, *VoxieOS* will automatically mount to the drive and show its contents. This is a quick way to view files. **Note: large file types may not be as performant when viewing on a USB drive compared to being viewed from the VX1's hard drive.**

VoxieOS.ini settings

The *VoxieOS.ini* file is a user editable file for setting up the *VoxieOS* experience. This file is included along with the *VoxieOS.exe* by default and is installed under 'C:\Voxon\Software\VoxieOS\'.

When *VoxieOS* is launched it loads in the settings contained in this .ini file. Below is a table listing the values and parameters available.

Note: some of these options are also available on VoxieOS's menu tab.

Value (and default value)	Description
startpath=.\..\..\Media	Starting path: absolute or if relative, relative to VoxieOS.exe.
mediadir=.\	Location of VoxieOS's internal icons (models) relative to VoxieOS.exe. Best to keep these alongside VoxieOS.exe.
hide_exts=.c;.h;.ini;.kc;.asm;.bmp	Inaccessible file extensions (hide these file types for simplicity of browsing).
iconperrow=3;	How many icons per row. Range is 2 – 6.
showborder=0;	1=displays wireframe box around volume in some modes, 0=disable.
autocycle=0;	1=cycle through animations (ZIP)/heightmaps in current directory automatically, 0=disable
heimap_inv_ctrls=1;	1=invert rotation controls for heightmaps (control camera), 0=control model
model_max_fileng_preview=50000000;	Largest 3D model file size (in bytes) to show preview in icon. Larger files show as a generic icon.
heimap_max_fileng_preview=3000000;	Largest heightmap/image file size (in bytes) to show preview in icon. Larger files show as generic icon.
gobble_sphere_brightness=64;	For DICOM, 4 is default. Range:1-255. The size of the subtraction or addition ('gobble') sphere when changing the data.

VoxieOS Menu Settings

When *VoxieOS* isn't displaying media, the menu contains the following options:

Option (and default setting)	Description
Sort By: Name	Sort the media by name or by media type.
Order : Asc.	Sort the media in Descending or Ascending order.
Refresh Listing	Refreshes the current directory. If files within the directory have changed the changes will be updated in the file browser.
Load All pos&ori	Loads in all the pos&ori settings within the current working directory.
Save All pos&ori	Saves all the pos&ori settings within the current working directory.
Icons per line : 3	How many icons / files are displayed per line in the file browser.
Show borders: off	Toggles between showing a white border around the volume.
Save VoxieOS.ini	Save changes to *voxieOS.ini*.

Quitting VoxieOS

Unlike most VX apps, rather than just pressing 'ESC' to exit the application, SHIFT + ESC is required to exit *VoxieOS*. This is because VoxieOS is used to launch other VX Apps and binding ESC to exit *VoxieOS* could result in accidentally closing *VoxieOS* as well as the other VX app.

Saving Rotation, Detail, Orientation and Scale Settings (Pos&Ori)

Each media file can have its transform and view settings saved. These settings are collectively known as 'pos&ori' settings, containing the position, orientation, scale, color, gamma, density and shading settings for that particular media. The 'pos&ori' settings are stored in the media's local folder within the *posori.ini* file.

When a media type is opened, *VoxieOS* will load up the 'pos&ori' settings and show the media as instructed by the *posori.ini* file.

Any 'pos&ori' settings made will be remembered until *VoxieOS* is closed. 'Pos&ori' settings can be saved, loaded or reset using the buttons found within the viewing media type's unique menu tab.

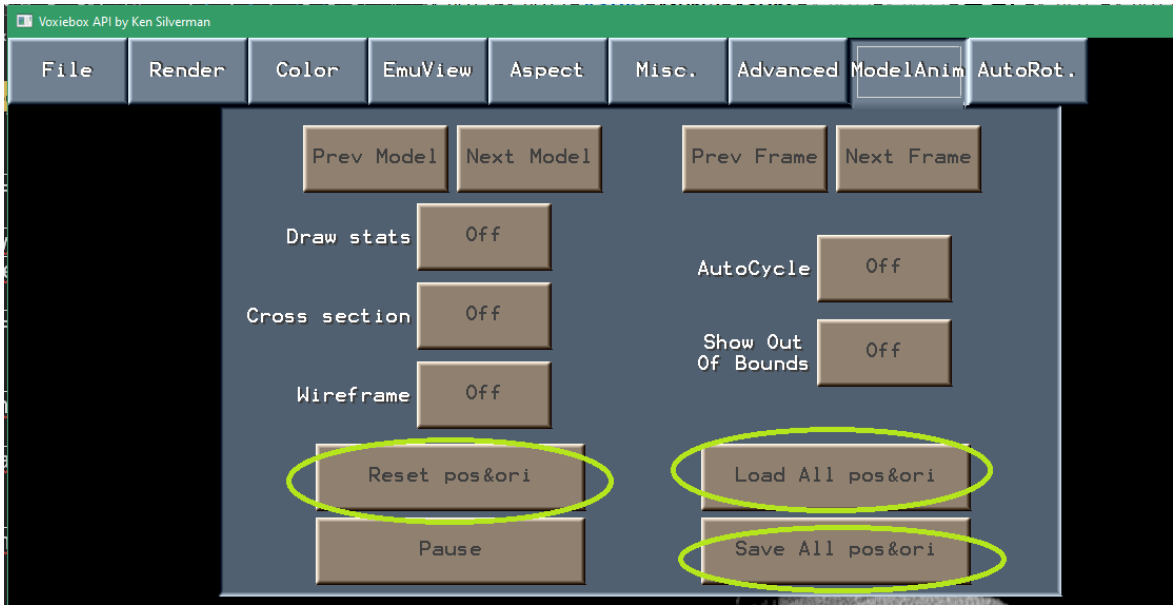

Figure 3 Model / Animation unique menu tab with pos&ori settings

Media Examples Included in the Developers Kit

The *Developers Kit* contains a variety of media examples. Most of the general media examples are located within the Media folder ('C:\Voxon\Media\Media Examples'). More advanced media types and tools can be found under 'C:\Voxon\Developers\Media Creation'.

Filename	File Location	Description
Muscle_Car.obj	Media Examples \ Models \ OBJ \ MuscleCar	Obj with material file but no texture files. (colours from the material file).
Apricot_02_hi_poly.obj	Media Examples \ Models \ OBJ \ Apricot	Obj with texture.
Ant.stl	Media Examples \ Models \ STL	STL image.
Stuart.kv6	Media Examples \ Models \ KV6	Kv6 example.
Fish.zip	Media Examples \ Animations	Animation Sequence.
Human_hand.zip	Media Examples \ DICOM	DICOM example.
Bun_zipper,ply	Media Examples \ Models \ Ply \ Stanford Bunny \ reconstruction (mesh)	Ply mesh example.
Bun000.ply	Media Examples \ Models \ Ply \	Ply point cloud example.

	Stanford Bunny \ data (pointcloud)	
Acetone.mol	Media Examples \ Molecules	Mol example.
Earth1k.jpg	Media Examples \ HeightMap	HeightMap with texture.
Canyon.jgp	Media Examples \ HeightMap	HeightMap without texture.
PolyCount_test	Developers Kit \ Media Creation \ PolyCount Test	Resolution test.
Fishhook.zip	Media Examples \ Thumbnail Example	Example for a custom thumbnail for animation.

Viewing 3D Models with VoxieOS

VoxieOS can view many types of 3D formats such as .OBJ, .STL, .PLY, .KV6. To view a model, navigate to it through the file browser and press 'Enter' on the keyboard, or click the left button on the Space Mouse or mouse. *VoxieOS* will load the media when viewing these file types. The model can be moved in any direction, rotated by any angle and scaled to any size. Models can also be rendered to show either their surfaces or just as a wireframe.

Whenever *VoxieOS* is viewing a media file, a unique menu tab appears on the VoxieMenu and specific key bindings are displayed on the secondary (touch) screen.

These are the Voxie Menu items when viewing a 3D Object

Option	**Description**
Prev Model	Loads and views the previous model in the current directory.
Next Model	Loads the next model in the current directory.
Prev Frame	If the model is part of an animation, shows the previous frame
Next Frame	If the model is part of an animation, shows the next frame
Draw Stats	Displays frame number and the filename of the model on the volumetric display.
AutoCycle	If enabled, when viewing a model after a short period of time the next model in sequence will be loaded. This process will repeat and loop from the last model back to the first creating an idle 'slideshow' experience.
Cross section	Views only a cross section of the model. The cross section can be adjusted by moving the transform.

Show Out Of Bounds	If the model is outside of the volumetric display's bounds, an arrow on the display will point to where the model is located.
Wireframe	If enabled, the current model will be shown as a wireframe model (surfaces are shown otherwise).
Reset pos&ori	Resets the current model's pospori to default
Load All pos&ori	Loads all the pos&ori settings for all the models in the current working directory.
Pause / Resume	Pauses the current animation.
Save All pos&ori	Saves all pos&ori settings for models within the current working directory.

Basic Summary of Supported Model Format Types

.OBJ (Wavefront .obj) "Object" File

OBJ files only store geometry data and reference additional material files (.mtl) which in turn can reference additional texture files (raster images PNG, JPG etc). as OBJ filetype relies on referencing other files aren't as straightforward as the other media types to use.

The .obj and .mtl files are human readable and can be edited with a text editor. Inside them you'll find references to the texture file names. Sometimes these file references are wrong and need to be edited for the materials and textures to load.

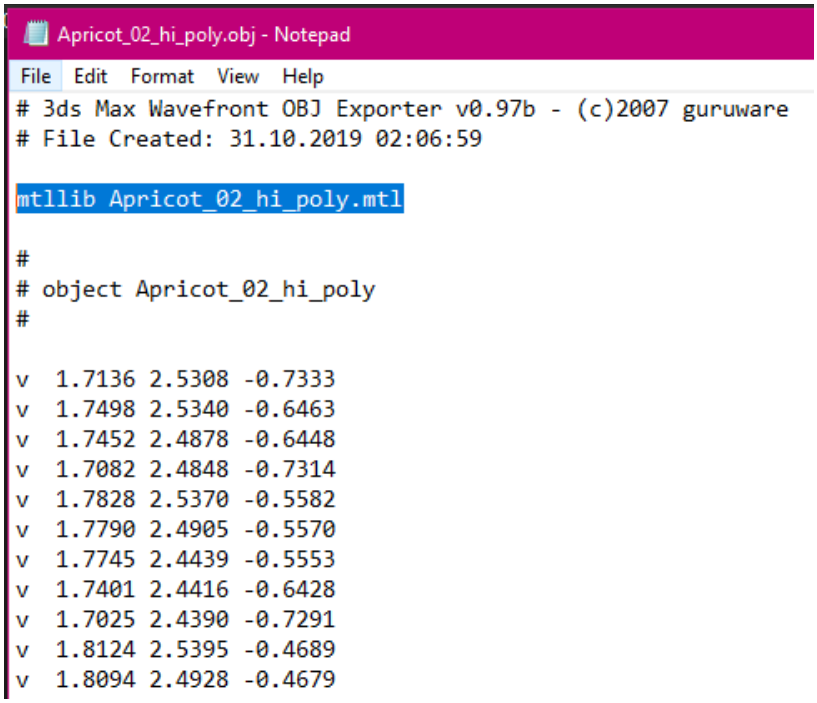

Figure 4 The contents of the Apricot_02_hi_poly.obj

The Apricot media example .obj file is very long but at the top it is referencing the material file. Inside the file you can see all the data defined.

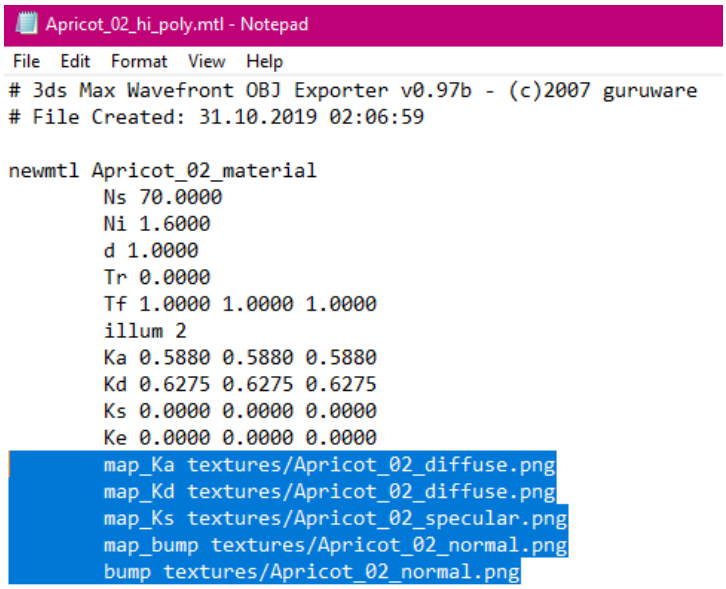

Figure 5 The contents of the Apricot_02_hi_poly.mtl material file

All the values for the material are defined here and any textures are referenced here too. As you can notice there are some PNG images that are holding the texture data.

KV6 / KVS format

A KV6 is a voxel based format developed by Ken Silverman. Instead of using a mesh based on geometry data, the model is drawn from a series of voxels (a 3D pixel). If you zoom in closely to any KV6 model you'll notice the voxels. Textures and materials are embedded into the voxels as each voxel has its own colour. The KV6 file doesn't need any additional files. The main drawback to a KV6 file is that because it is a voxel based format and not geometry it can't be scaled (zoomed in) without losing resolution (unlike as a vector based format).

Included in the *Developers Kit* is *Poly2vox*, a command line application that can convert many 3D model formats into a .kv6. *Poly2Vox* can be located at '…\Developers Kit\Media Creation\Media Creation Tools\'

Running 'Poly2Vox /?' In a command prompt will show the parameters and syntax for using *Poly2Vox*.

KVS file format is similar to a KV6 but uses uses an octree for better compression You can convert KV6 to KVS using:

"pnd3d caco.kv6 /out:caco.kvs"

Pnd3D is a voxel engine that Ken Silverman created in the late 2000s as well as being its own Voxel engine it can convert .KV6 into .KVS

Pnd3D.zip is located in '…\Developers Kit\Media Creation\Media Creation Tools\'

```
POLY2VOX [input] [output] [/v#] [/s#] [/f#] [/n#] [/r#] [/m#] [/x#] [/p(file)]
by Ken Silverman (http://advsys.net/ken)  Compiled: Sep  7 2017

Converts models from polygon to voxel format.
Supported polygon formats: ASC,3DS,MD2,MD3,OBJ,STL
Supported   voxel formats: VOX,KVX,KV6,VXL (default:KV6)
Supported texture formats: PNG,JPG,TGA,GIF,CEL,PCX,BMP,DDS
POLY2VOX can load files out of a ZIP file.

 /v#  Specify voxel size of longest dimension. 1-1024, <=256 for KVX
 /s#  Specify explicit scale factor. Use this to ensure the size of all frames
      is consistent. This factor depends on the coordinate system used by the
      polygon model, so it can be anything. Run without the scale factor first
      to find a reasonble starting value to try.
 /f#  Specify frame number (MD2/MD3 only)
 /n#  Specify next frame number for interpolation (MD2/MD3 only)
 /l#  Specify texture interpolation method: {1:nearest, 4:4x4 (default)}
 /r#  Specify frame interpolation ratio: {0.0-1.0}, default:0.0 (MD2/MD3 only)
 /m#  Specify number of mips to save: 1,5, default:5. (KVX only)
 /k?  Specify illumination model for OBJ (Ex: /ka, /kd, /ks, /ke)
 /y   Polygon render (default)  /y2: Polygon render using supercover
 /w   Wireframe render          /w2: Wireframe render using supercover
 /x   Experimental xor-style render for gap-less models;buggy color conversion
 /c   Center model in bounding box (default is to use polygon file's 0,0,0)
 /cm  Center of model is centroid
 /b(l/r/b/f/u/d)# Clip boundary (Left/Right/Back/Front/Up/Down). Ex: /bu-1.2
 /t(file)  Select a texture file (if not specified in polygon model).
 /p(file)  Specify Build-style palette (first 768 bytes of file, range:0-63)
 /z(file)  Specify a ZIP file to mount. Files inside seen as local dir.

Examples:
 poly2vox bike                        (finds bike.*, writes bike.kvx, size=128)
 poly2vox bike.3ds bike.kv6 /v250   (reads bike.3ds, writes bike.kv6, size=250)
 poly2vox land land.vxl        (finds land.*, writes land.vxl, size=1024^2*256)
 poly2vox pig.md2 pig.kvx /v128 /f2 /n3 /r.5       (convert interpolated frame)
 poly2vox trooper.md2 trooper.kvx /s.115 /f0        (use explicit scale factor)
 poly2vox trooper.md2 trooper.kvx /s0.115 /f0 /ppalette.dat      (user palette)
 for /L %i in (0,1,50) do start /i poly2vox /zmonst.zip monst%i.obj     (batch)
```

Figure 6 The embedded help file for Poly2Vox for syntax help.

STL "Standard Triangle Language" file format

STL files are commonly used for 3D printing and only contain the geometry data. There is no material, texture, or color data stored in an STL. STL files also can be in ASCII (human readable) or binary. STLs are great for 3D models that don't require the use of colour or materials.

PLY "Polygon" file format

PLY was originally developed to store data for 3D scanners. The PLY format can store geometry (mesh) or point cloud data. The point cloud use case is of interest to Voxon's

technology as it is the only native point cloud based media type supported. The file can be in ASCII (human readable) or binary.

3D Model Sequence (Animation) with VoxieOS

Any collection of 3D models can be played as a sequence. For *VoxieOS* to detect an animation sequence the filenames need to be named sequentially (e.g animation1.obj,animation2.obj,animation3.obj,...). You can either compress these files into a .zip file. Or put them inside their own sub directory (folder).

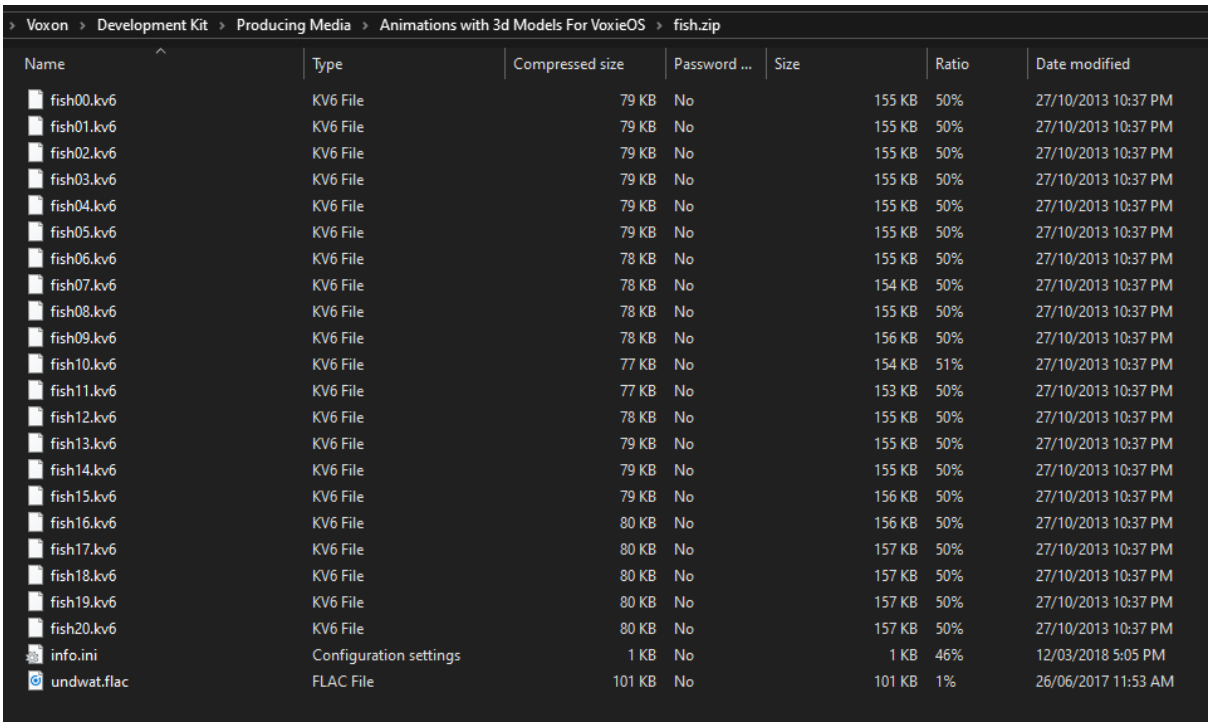

Figure 7 The contents of fish.zip animation sequence.

The contents inside the example animation *fish.zip* is a collection of 21 K6V fish models in sequence. The *info.ini* file is the playback properties and the undwat.flac is a sound file (optional) that accompanies the animation.

Tip for batch renaming files: a quick way to rename a whole heap of files at once is to select them all in Windows Explorer and right click and click rename.

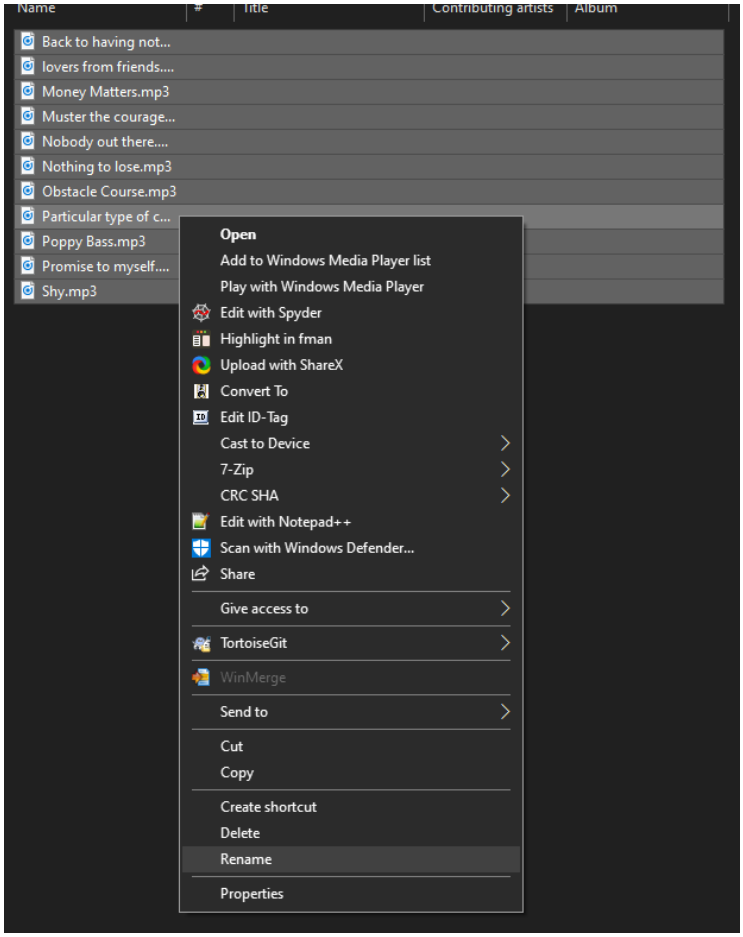

Figure 8 Selecting all the files in Windows Explorer and clicking Rename will batch rename all selected files

Rename the files and they will be renamed sequentially!

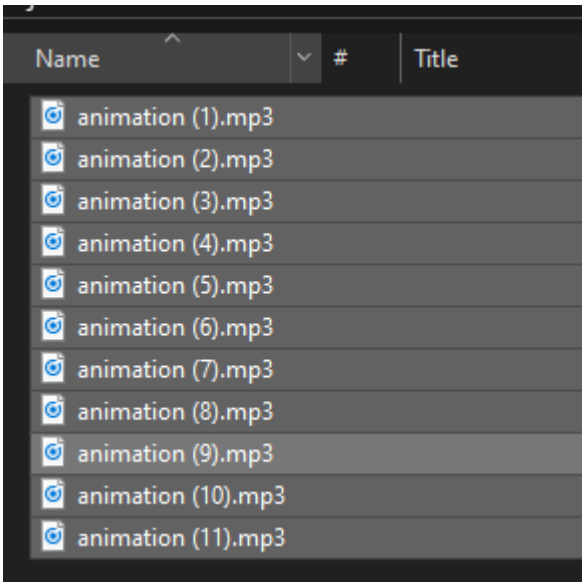

Figure 7 And presto! All the files have been renamed

Yes, I know that these are .mp3 files but it doesn't matter the extension!

Having the files in a .zip or just in their own folder will play just the same. While a .zip file does not need the *info.ini* however to use a folder as an animation the files must be accompanied with an *info.ini.*

The *info.ini* file contains the parameters for the animation playback.

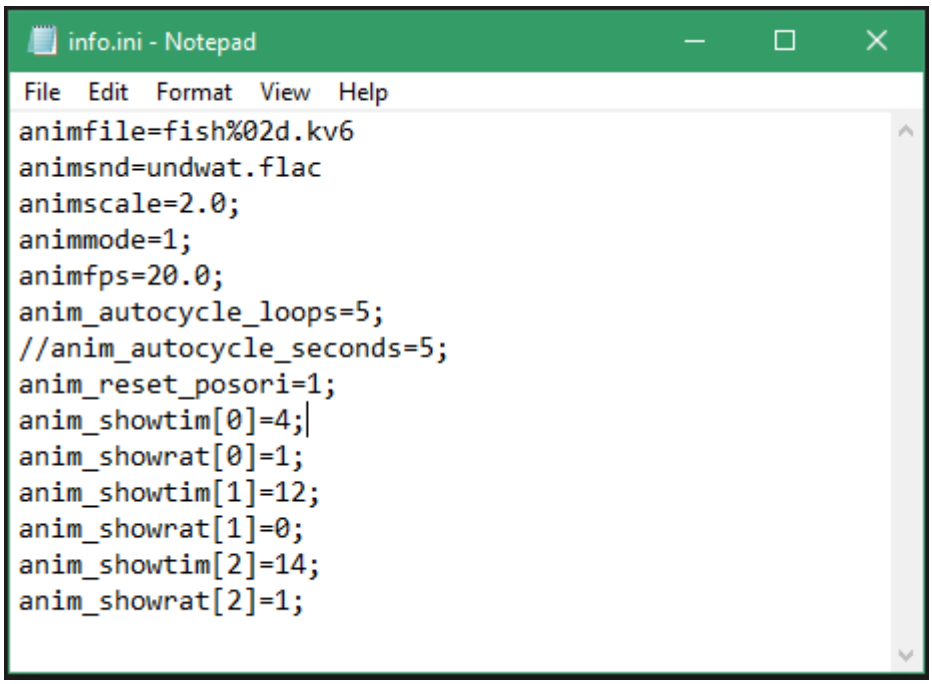

Figure 8 The contents of the info.ini

A detailed description of the values and parameters available in an *info.ini* file.

animfile=fish%02d.kv6	The filenames to open, limited printf-style formatting to select files in order. Typically: file%d.ext or file%0#d.ext
animsnd=undwat.flac	audio file to play .wav, .flac, .mp3
anim_staticfile=bulstand.kvs	Add a static 3D model to the animation
animscale=2.0;	playback at scale
animmode=1;	Animation playback mode 0=forward, 1=pingpong
animfps=20.0;	frames per second to play back at (default is 15)
animforcescale=1.0;	set a forced scale - useful when animation frames vary in size
anim_autocycle_loops=1;	in autocycle mode, tells how many loops to play before next file

anim_autocycle_seconds=5;	in autocycle mode, tells how long seconds to play before next file
anim_autocycle_nokillsound=0;	if enabled continues audio playback after changing animation
anim_reset_posori=1;	reset pos & ori on load of animation
anim_showtim[0]=0.5;	custom animation sequence. Time in seconds.
anim_showrat[0]=0.2;	ratio from 1st to last frame:{0.0..1.0}
anim_showtim[1]=1.5;	can have multiple animation sequences just add to the array anim_showtim[1], anim_showtim[2].. etc"
anim_showrat[1]=0.4;	ratio from 1st to last frame:{0.0..1.0} for this segment

Showing a Custom Thumbnail for Animations

A custom thumbnail can be shown on VoxieOS by including a 3D media file named as ‘thumbnail.*’ (* = the relevant file extension) within the zipped folder. Standard 3D files are supported; KV6, KVS, STL, PLY and OBJ. See the ‘Fishhook.zip’ example located in the “Media Examples’ folder (by Default C:\ Voxon \ Media \ Media Examples \ Thumbnail Example‘).

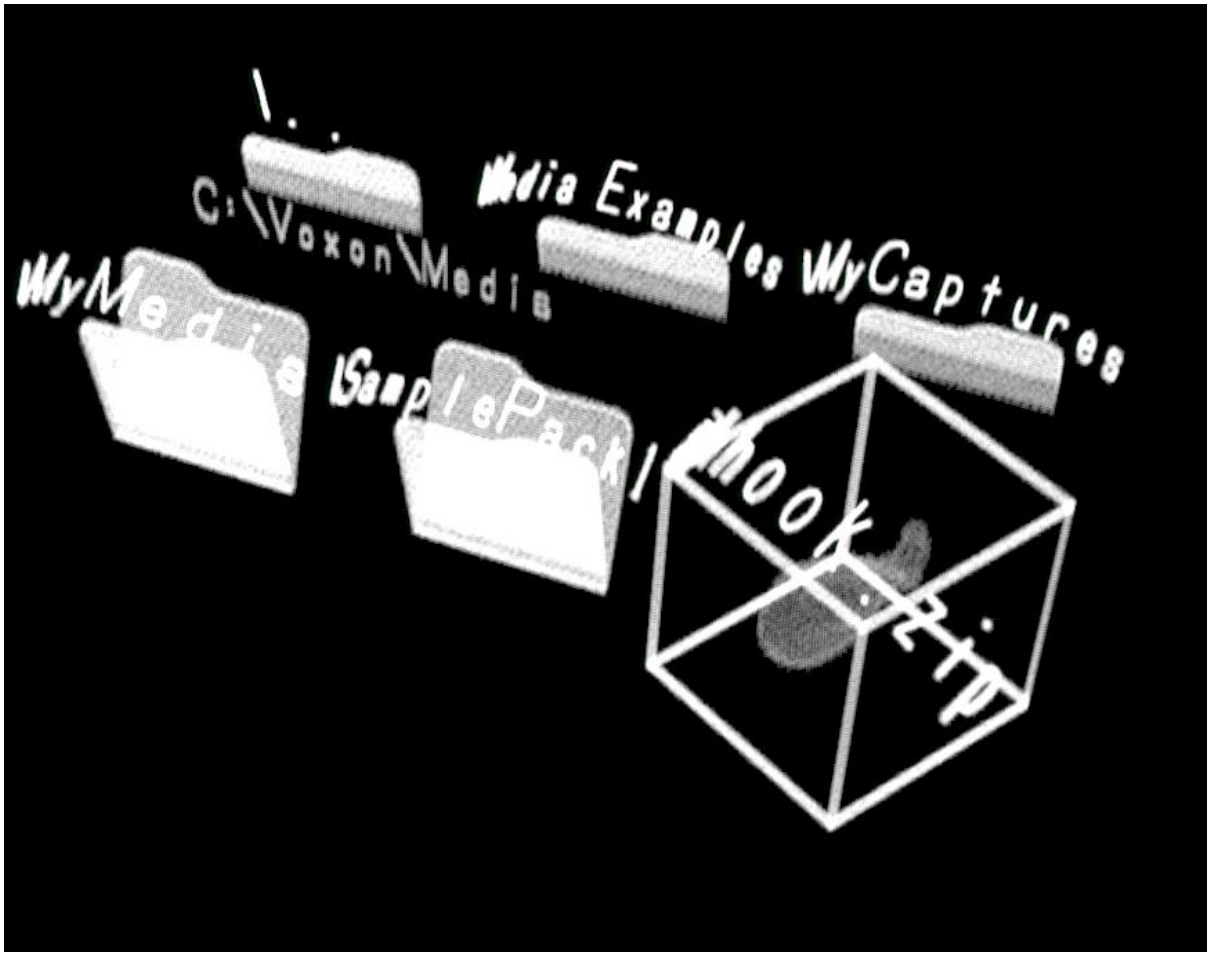

Figure 9 Custom thumbnail for animation. (loads thumbnail.KV6 from the embedded zip animation)

HeightMap with VoxieOS

Heightmaps are 2D Images with a height map channel next to it. These can be any 2D image types (JPG, PNG, GIF, BMP, PCX… etc) but require a 1:1 height channel to the right side of the image.

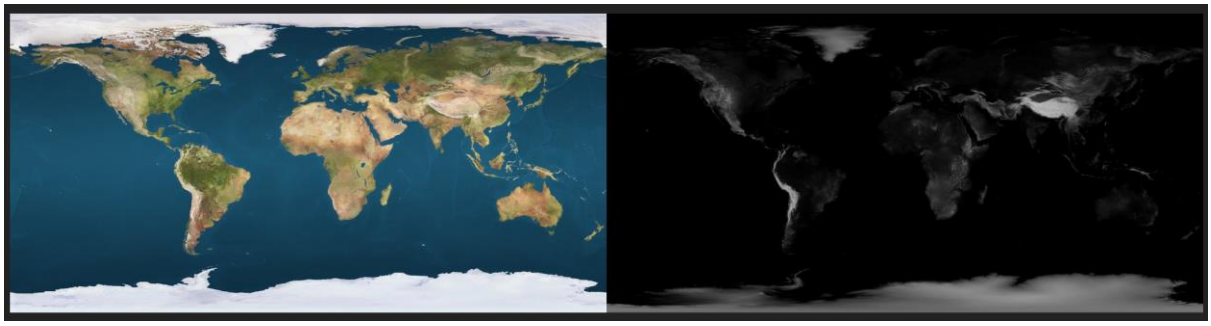

Figure 10 Viewing a heightmap in a standard 2D image viewer

Looking at the earth1k.jpg heightmap example in a 2D image viewer shows the image on the left and the heightmap on the right. The ‘heightmap’ part of the image works by white (color

0xffffff) representing the highest point of the map and black (0x000000) being the lowest point of the map and all the greyscale being the height in-between. Actually the left image can be considered the texture and the right image is the height map.

Viewing this file through *VoxieOS* will combine these two images to make a heightmap.

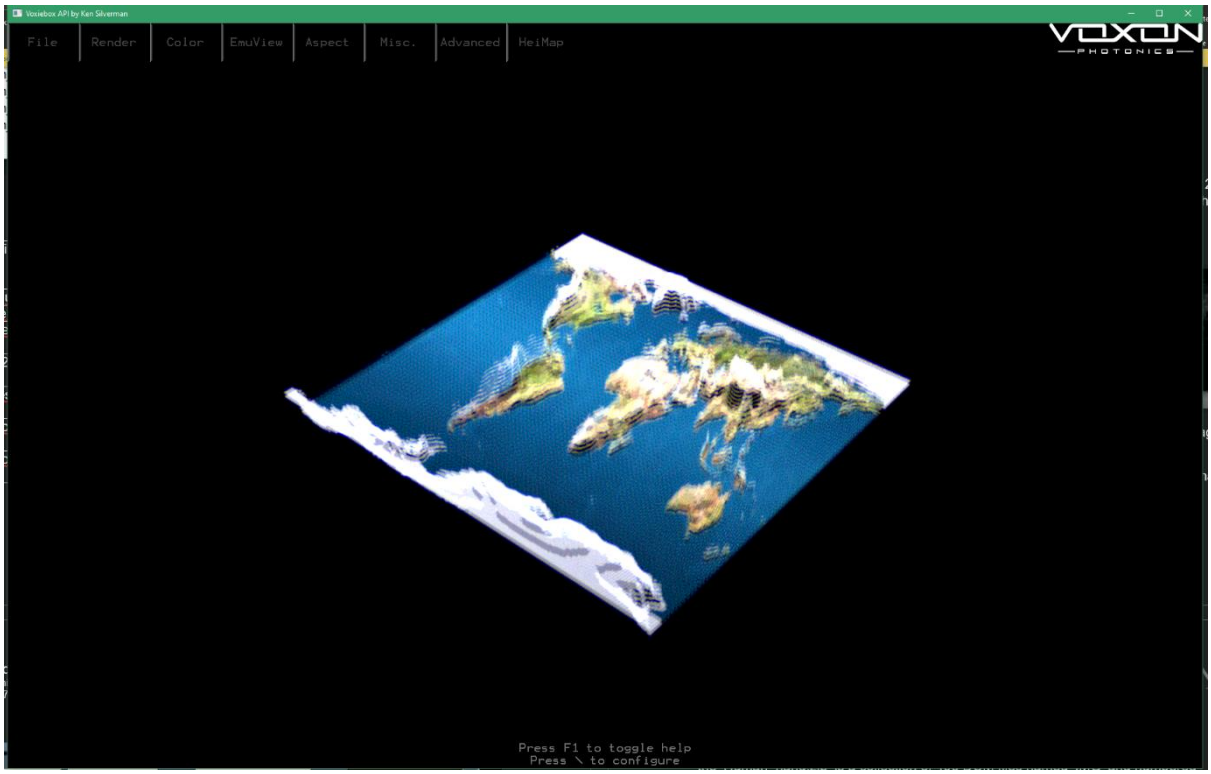

Figure 11 Viewing a heightmap on the Voxon simulator

Heightmaps cannot be transformed along the Z axis. They always sit 'flat' but their height's scale, the distance between the map's highest and lowest point can be adjusted.

Like any media type in *VoxieOS* Heightmap has its own set of options under its file menu. Here is a breakdown of their unique menu tab.

Option	**Description**
Prev Model	Loads and views the previous model in the current directory.
Next Model	Loads the view of the next model in the current directory.
Invert hgt=Off	Inverts the heightmap. Higher density areas are displayed lower on the display while lower density areas are displayed higher on the display.
Invert ctrls=On	Inverts the rotational controls (rotate object VS rotate camera)
Height scaling=1	How extreme the high
AutoCycle=Off	If enabled when viewing a model after a short period of time the next model in sequence will be loaded. This process will

	repeat and loop from the last model back to the first creating an idle ‘slideshow’ experience.
Slice Dither=On	Add Dither to the slices. Makes the vertical slices appear more smooth
Text Filter = Bilinear	Switch between Bilinear Interpolation or nearest neighbour. Bilinear creates smooth transitions between the heightmap’s points. Nearest Neighbour culls the data to a point and creates a more jagged appearance.
Texture=On	Toggles on and off the texture for the heightmap (the 2D image)
Reset cam	Reset the viewpoint to default
Load All pos&ori	Loads all the pos&ori settings for all the models in the current working directory
Save All pos&ori	Saves all pos&ori settings for models within the current working directory.

Heightmap with no texture

It's possible to view a heightmap without a texture. Canyon.png is an example of this. Here is what the file looks like on a typical 2D image viewer.

Figure 12 Viewing canyon.png in a standard 2D image viewer.

When viewing *canyon.jpg* on the volumetric display (or simulator) it will be interpreted as a heightmap.

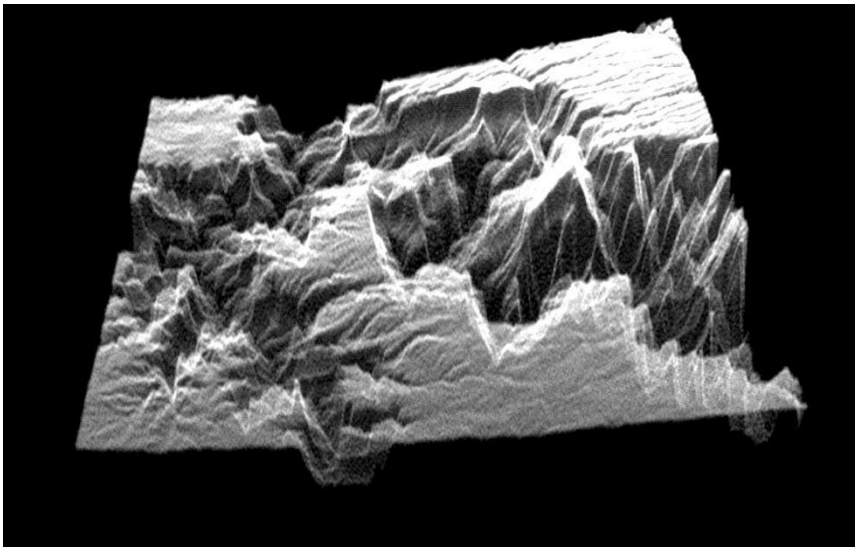

Figure 13 Viewing canyon.png on the Voxon Simulator.

DICOM / DICOS with VoxieOS

Voxon's Photonic Engine has native support for viewing DICOM or DICOS data. DICOM stands for Digital Imaging and Communications in Medicine (DICOS is the security equivalent). This is a powerful data type that shows medical or security imaging. For more information about the DICOM data type can be obtained from https://www.dicomstandard.org/

For the *VoxieOS* to view DICOM/DICOS files (files with the. DCM , .DCS extension), they need to be compressed in sequence into a single file aka a 'compressed .zip file' or placed together within a directory. Like an animation sequence these files need to be named sequentially though there is no need for an info.ini to be included.

For DICOM / DICOS files to have any useful depth you need many DCM files. A DCM file is a single slice of the image. A collection of at least 50 slices of your image is needed to make it look interesting on our volumetric technology. The human hand example (*human_hand.zip*) has about 500 DCM files.

While DICOM files can be in a few different file types. They can be converted to DCM by an open source program 3D Slicer (https://www.slicer.org/). 3D Slicer is great for preparing DICOM files. In fact, any stack of raster images can be opened with 3D Slicer and converted to DCM. This is a great way to use the DICOM media file type for non-medical uses.

Showing a Custom Thumbnail for DICOM

A custom thumbnail can be shown on VoxieOS by including a 3D media file named as 'thumbnail.*' (* = the relevant file extension) within the zipped folder. Standard 3D files are supported; KV6, KVS, STL, PLY and OBJ. See the 'Fishhook.zip' example located in the

"Media Examples' folder (by Default C:\ Voxon \ Media \ Media Examples \ Thumbnail Example').

Use the 'Save to STL' feature (located on the DICOM tab within VoxieMenu). To create a good 3D model of DICOM.

Where to find DICOM files

Any medical image company will know of the DICOM file type. Often due to issues of privacy, these files are not accessible by the public.

The Cancer Imaging Archive is a website which allows you to download free anonymous DICOM data which you can use to explore this media type.
https://www.cancerimagingarchive.net/

Video Tutorial of Making a DICOM file

A video tutorial of viewing DICOM files on a VX1 can be found here: https://www.youtube.com/watch?v=bOfc4GKFcXQ&t=17s it is also included in the Developers Kit and can be found at '…\Voxon\Developers Kit\Media Creation\DICOM'

DICOM Media Example – Human_hand.zip

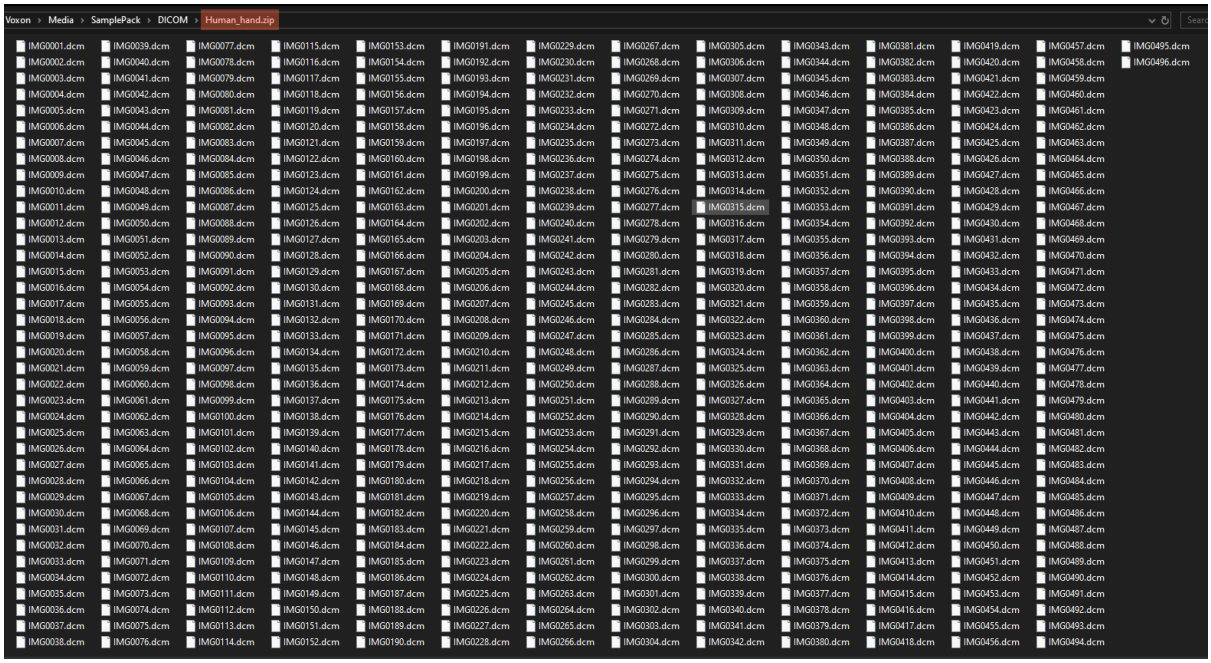

Figure 14 The contents of the human_hand.zip example DICOM file

The *human_hand.zip* is a collection of 496 DCM files named and numbered sequentially. Each DCM file is one slice of the image.

When loading ‘Human_hand.zip’ via VoxieOS

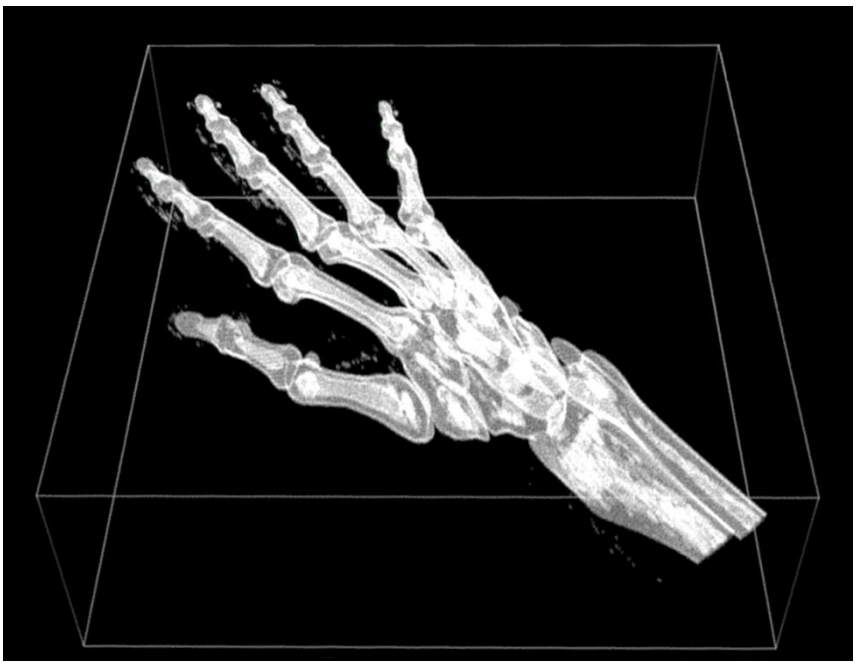

Figure 15 Viewing the ‘Human_hand.zip’ DICOM with the Voxon Simulator

Viewing a DICOM/DICOS file

To render a DICOM image, a contour value needs to be set. A DCM image is like a heightmap; it shows the density of the image. The whiter the pixel the higher the density of the material. The contour value determines a threshold of what density to render and what to not show. Parsing through the contour thresholds will reveal different data within the DICOM image. As skin and bone have different densities; it’s easy to isolate either the skeleton, or the skin by adjusting the contour value. DICOM data is so rich that additional tools and processes are usually needed to help interpret the data.

Figure 16 adjusting the contour thresholds for viewing a DICOM

In this example, the contour threshold for the purple channel is set high to reveal only the skeleton of the hand. The contour threshold of the blue channel is set low to reveal only the skin of the hand.

The contour value is a powerful way to parse through DICOM data and render what is important.

There are many options within the DICOM's unique menu tab. Here is a description of each attribute.

Option	**Description**
Prev Dicom	Loads and renders the previous DICOM zip image in the current directory.
Next Dicom	Loads and renders the next DICOM zip image in the current directory.
Detail Slider	How much detail to show in the image. Higher detail means more voxels are being rendered, reducing system performance.
Detail@Stop	If enabled when the image is not being adjusted (moved, rotated or scaled) the image will show at high detail.
Contour Value 1	The threshold for the 1st contour value. If you see no image slide this value around

	until an image is present. The color blocks beneath the slider represent what color to render the data.
Contour Value 2	The threshold for the 2nd contour value. If you see no image slide this value around until an image is present. The color blocks beneath the slider represent what color to render the data.
Reset pos&ori	Resets the current pos&ori for the image
Load All pos&ori	Loads all the pos&ori settings into memory
Cross section	If enabled shows a cross section of the DICOM
Draw Stats	Shows debugging and extra stats on the secondary (touch) screen.
Save To STL	Exports a STL model of the current image. The file is named 'Voxie000.stl' and located where the VoxieOS.exe file is stored.
Save to STL;flip	Exports a flipped STL image of the current image. The file is named 'Voxie000.stl' and located where the VoxieOS.exe file is stored.
Save All pos&ori	Saves al the pos&ori settings made within the current working directory to posori.ini files
Wire frame	If enabled shows the DICOM as a wireframe instead of a surface
Ruler	Enables a ruler for accurate (assuming the meta data is correct) measurement Pt 1 – define the 1st point Pt 2 – defines the 2nd point. Revealing the distance between the two points Pt 3 – shows the angle of the line from another point

'Gobble Mode' with DICOM

Pressing the 'G' key while viewing a DICOM will enter 'gobble mode'. While in gobble mode pressing the left button on the mouse or space mouse will erase all the data within the sphere. The diameter of the 'gobble' can be adjusted by holding the right mouse or space mouse button. Gobble mode can be useful to remove unwanted data to increase clarity of the image. Pressing 'G' again when in gobble mode will exit gobble mode.

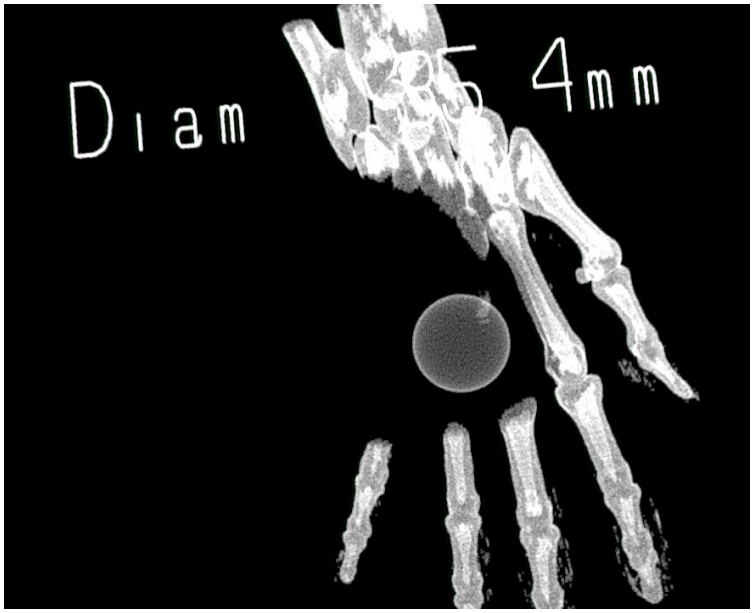

Figure 17 'Gobble mode' in action.

Similar to 'Gobble Mode' pressing the 'F' key will enter 'fill mode' where the sphere can be filled with data instead. Pressing 'F' a second time will exit 'fill mode'.

Molecule File Type

VoxieOS can read a .mol file type which is created using the MDL molfile format, a chemical file format. A MOL file contains plain text information about atoms and bonds. Viewing molecules in true 3D can make them easier to understand.

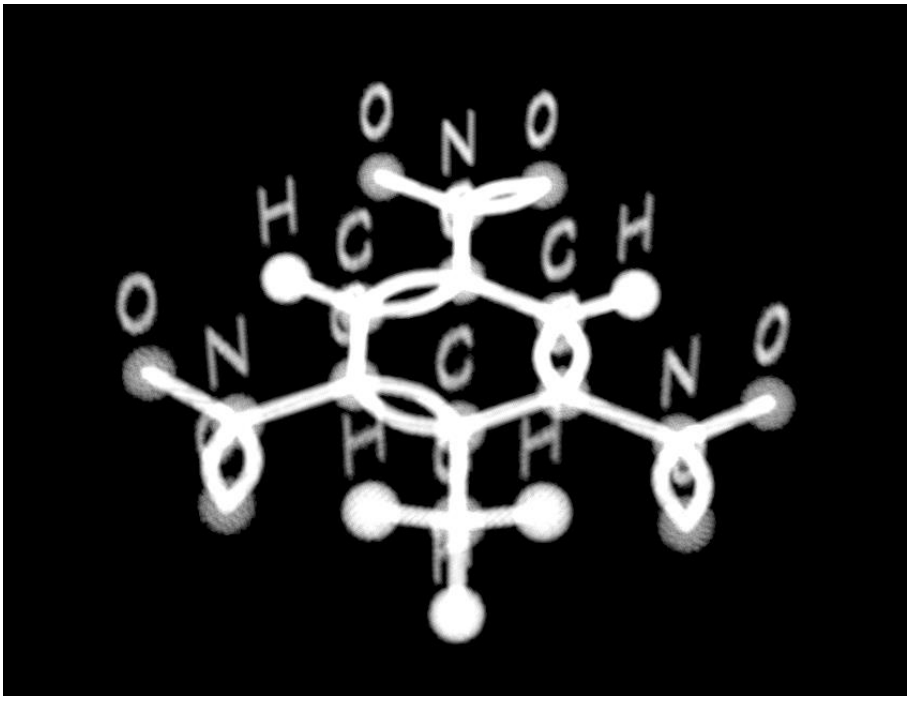

Figure 18 Viewing a .mol file with VoxieOS.

When viewing a .mol file the following unique options are available.

Option	**Description**
Prev .MOL	Loads and renders the previous .mol file in the current working directory.
Next .MOL	Loads and renders the next .mol file in the current working directory.
Show symbols	If enabled, it shows the chemical symbol of the node.
Node size	How large to render a node.
Bong thickness	How thick the bonds are rendered.
Reset pos&ori	Resets the pos&ori values to default.
Load All pos&ori	Loads all the pos&ori values from the current working directory's pos&ori.ini.
Pause	pauses or animates the model.
Save All pos&ori	Saves all the pos&ori values to the pos&ori.ini file in the current working directory.

Playing Depth Camera Footage with VoxieOS

Figure 19 Playing depth camera footage (.kni file) through VoxieOS

Depth camera footage captured with *KniView* or *Knife* (.kni file format) can be played back within *VoxieOS*. This is footage that has been captured by a stereoscopic infrared camera. A KNI file stands for 'Ken's Natural Interaction' The format was developed by Voxon's Ken Silverman. It is a volumetric interpretation of depth camera footage.

Depth camera footage is created by combining various types of image data to create a depth image. A depth image knows the relative (or absolute) distance between each pixel so that an image could be created in a 3D space. The image is never fully 3D as it is only from the camera(s) perspective. To achieve this, each depth camera model works slightly differently but they usually comprise three layers of data. Firstly a depth layer, usually achieved by 'time of flight', or from interpreting polygraph views. The other two layers are formed by traditional RGB colour camera(s) and an infrared camera(s). Depth camera hardware usually contains at least one microphone to record audio.

.kni playback can be moved, rotated and scaled similar to other types of media and *VoxieOS* stores their own pos&ori settings. When playing back a .kni file, you can tweak the gamma and density settings to really bring out the quality of the image.

Option	**Description**
Prev Kni	Loads and views the previous Kni video in the current directory.

Next Kni	Loads and views the next Kni video in the current directory.
Play Speed = 1	The playback speed. 1 is the default Realtime playback speed
RGB / Infrared	View the RGB or Infrared channel
AutoCycle	If enabled and within a folder of multiple KNI views the content will cycle to the next KNI file after playback is ended
Reset pos&ori	Resets the current pos&ori settings for the current KNI video
Load All pos&ori	Loads the pos&ori.ini values saved in the current working directory.
Pause	Pauses the KNI playback
Save All pos&ori	Saves the pos&ori settings into the local pos&ori.ini file.

Using a Depth Camera with Voxon Display Technology

Voxon's volumetric technology is able to capture, stream, record and playback depth camera footage. For this to occur, a compatible depth camera is required. Supported models include:

- Microsoft Kinect V2
- Microsoft Azure Kinect
- Creative/Intel SR300
- Intel D415

Voxon recommends a Microsoft Azure Kinect or Microsoft Kinect v2.

The software needed to do this is located within 'C:\Voxon\Software\KniView'. The two main programs are *KniView*, and *Knife*. Both programs will assume a recording is required.

Note: When launching either application they will launch in live video capture mode and require an attached depth camera. To run them in a playback mode they must be run as a command line with a filename to the kni media to be playback.

KniView – playing live footage or recording on a volumetric display

KniView is a VX app (that can also be run as a standard Windows application) used for playing back or streaming depth camera content. *KniView* can have a network connection to other systems running *KniView* for 'volumetric video calls'. TCP or UDP connections are available. Networked systems do not need to send their own footage and *KniView* does not need to run on actual Voxon hardware. Multiple connections are available too. *KniView* can play KNI files but they can only be loaded via command prompt.

Figure 20 KniView running on a Voxon VX1

KniView streaming live video content using a Microsoft Azure Kinect camera.

Knife – recording and editing depth camera

Knife is not a VX app. It is a standard Windows application designed to record and edit depth camera footage.

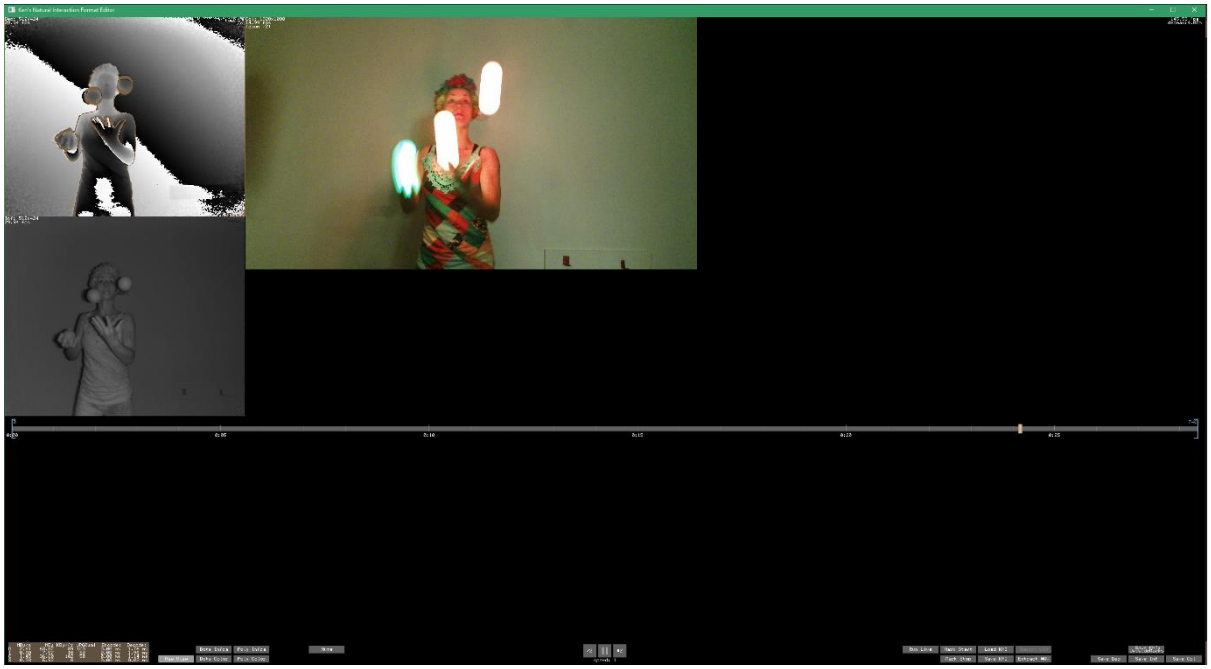

Figure 21 Knife being run on a Windows PC

Inside the *KniView* folder are many files. Most of the included .dlls are libraries required to access the different camera types, as each camera type has been reverse engineered to work with the .knii file type. Here is a summary of the important files for developing content:

File	Description
Kni_newwav.exe	Command line application to replace a Kni file's embedded wav file. Run as a command prompt "Kni_newwav KniFileToEdit.kni wavefiletoEncode.wav 0" (the offset in seconds to encode the wave). The output will always be called 'temp.kni' so make sure you rename your new file.
Knife.exe	The executable for the Knife application. For recording, editing, and playing back KNI files on the standard Windows application.
Knife.txt	Help file for using Knife.
Knife_azure.exe	Knife application dedicated to run with Microsoft Azure camera.
Knife_intel.exe	Knife application dedicated to run with Microsoft Intel based cameras.
Knife_kinect.exe	Knife application dedicated to run with Microsoft Kinect cameras.
Knife_openNI2.exe	Knife application dedicated to run with OpenNI2 support (Supports a variety of depth cameras).
Kniview.ini	User editable settings for Kniview.
KniView.txt	Help file for using KniView.
Juggler.kni (located under the media sub directory)	Example .Kni media file.

Using Kniview

KniView is a VX application for live streaming or playing back pre-recorded depth camera footage. To live stream depth camera footage, a compatible depth camera is required.

KniView can operate in three modes:

1) Playback live depth camera footage. (it can't record capture)
2) Playing back pre-recorded depth camera footage (.kni files)
3) Streaming a live depth camera through a network with other *KniView* applications Via TCP or UDP network connections.

The *KniView.txt* included in the *KniView* folder details more instructions on how to use *KniView*.

Up to 4 cameras can be streamed at once.

The .kni file type is made up of the following elements:

- Calibration parameters stored in camera's firmware (reverse-engineered from the libraries)
- Custom-written lossless compression for depth buffer (approximately 3:1 ratio)
- JPEG-compliant writer for Infrared (qual=80) and RGB frames (qual=50)
- audio - uncompressed

Launching KniView via command line

KniView is a command line application. **Running just *KniView* without any parameters will launch *KniView* as a Windows app in live capture mode.**

To access most of *KniView's* features, command line parameters needed to be defined when the program launches. Many settings including playback recorded captures, load onto the volumetric display, set which camera type and many other settings are all enabled via the command line.

To launch *KniView* with command line parameters, open a windows command line (press 'Start' on Windows and type 'cmd') and navigate to the *KniView* directory (by default C:\Voxon\Software\Kniview). **Typing in 'KniView media/juggler.kni' will play the juggler example video as a standard Windows application**

Figure 22 Command line for running KniView in playback mode as a Windows application

If you are running on Voxon hardware and want KniView to run as a VX app and on the volumetric display, the /v parameter is needed. **Typing in 'KniView media/juggler.kni /v' will play the juggler example video as a VX application on the volumetric display.**

Figure 23 Command line for running KniView in playback mode as a VX application

Note: If no .kni media file is specified, *KniView* will run live footage from the connected camera.

Here is a list of all the parameters for *KniView*

Parameter	Description
/o	select input device: OpenNI2 library
/k	select input device: Microsoft Kinect v2
/r	RealSense 2 (SR300, D415, ..)
/z	select input device: Microsoft Azure
/w	select output device: render in a 2d window (default)
/v	select output device: enable Voxiebox mode (render onto volumetric hardware
/3	select output device: enable Nvidia 3DVision mode
/ir	select infrared instead of RGB color (for live OpenNI2)
/a	specify WAV/FLAC file to play
/b	specify brightness offset (0 .. 10)
/t	specify number of CPU threads to use
/d#x#(x#)	specify depth buffer resolution and fps (optional). Examples: /d1280x720x30 or /d512x424
/c#x#(x#)	specify color buffer resolution and fps (optional). Examples: /c1920x1080x30 or /c640x480
/H	host TCP connection
/J	[ip address or server name of TCP host]: connect to TCP host (must specify ip). Examples: /J:192.168.1.100 or /J:localhost
/h	host UDP connection
/j	[ip address or server name of UDP host]: connect to UDP host (must specify ip). Examples: /j:192.168.1.100 or /j:localhost
/p	override TCP or UDP port. Example /p:32123

Live Depth Camera

By Default, running *KniView.exe* with no parameters will put *KniView* in live depth camera view and display what the connected depth camera is viewing if it is outputting on the volumetric display (/v parameter). The transform of the footage can be adjusted (in a similar fashion to any other media type). The keyboard, mouse and space mouse can be used to rotate, move and scale the footage.

Note: When you first run KniView the camera's output footage may not be within the volume. You may need to move and adjust the transform to fit the volume. Enabling head tracking can be a quick way to see something on the volume.

Playing pre-recorded KNI format depth camera footage

Playing pre-recorded .kni media can be done via the command line. The .kni file needs to be specified when launching *KniView*.

Running "KniView.exe exampleCapture.kni /v" will play back the *exampleCapture.kni* file onto the volumetric display.

The pre-recorded media will loop once it is finished. The footage can be adjusted (in a similar fashion to any other media type). The keyboard, mouse and space mouse can be used to rotate, move and scale the footage.

The pos&ori values can be saved by pressing the Left Ctrl + S key. The pos&ori values are stored within the *KniView.ini* file.

Networking KniView

KniView supports TCP or UDP connections. One instance of *KniView.exe* needs to be the host, the other instances need to join the hosts connection. The host specifies an IP address and port number. When a connection network has been established there are some connection stats written on the secondary (touch) screen. Press 1 – 4 on the keyboard to select which camera feed to edit.

Connection types, ports and IP settings can be changed within the application or set in *kniView.ini*. Inside the KniView settings file are more settings which can tweak the connection speed and quality. Make sure a firewall isn't stopping the connection and the port you nominate is valid.

In-app controls

Here is a list of valid In-app controls when using *KniView*

Keyboard or input	**Description**
ESC	quit
1-4	select camera
`	select all cameras
Arrows Key	Mode media forward/back/left/right
RCtrl	Move media up
Mouse	rotate camera (VX app mode: spin camera & walk forward)
LMB+Mouse	rotate camera free direction (VX app mode only)
RMB+Mouse	zoom camera (VX app mode only)
, and .	rotate around model horizontally
PGUP/PGDN	rotate around model vertically
/	reset camera
Numpad / and Numpad *	change zoom
A or Z	change zoom
Numpad 5	Reset zoom
L.Shift	hold for 16x slower movement
R.Shift	hold for 16x faster
H	toggle head tracking mode for current camera
P	pause video
M	mute audio
Numpad – or Numpad +	adjust maximum depth to crop (default:8192)
R	toggle raw video
Spacebar	toggle Color vs. IR (always uses Depth)

X	mirror image (VX app only)
Ctrl+S	save current pos&ori and update default settings to KNIVIEW.INI.

In-app menu options for KniView

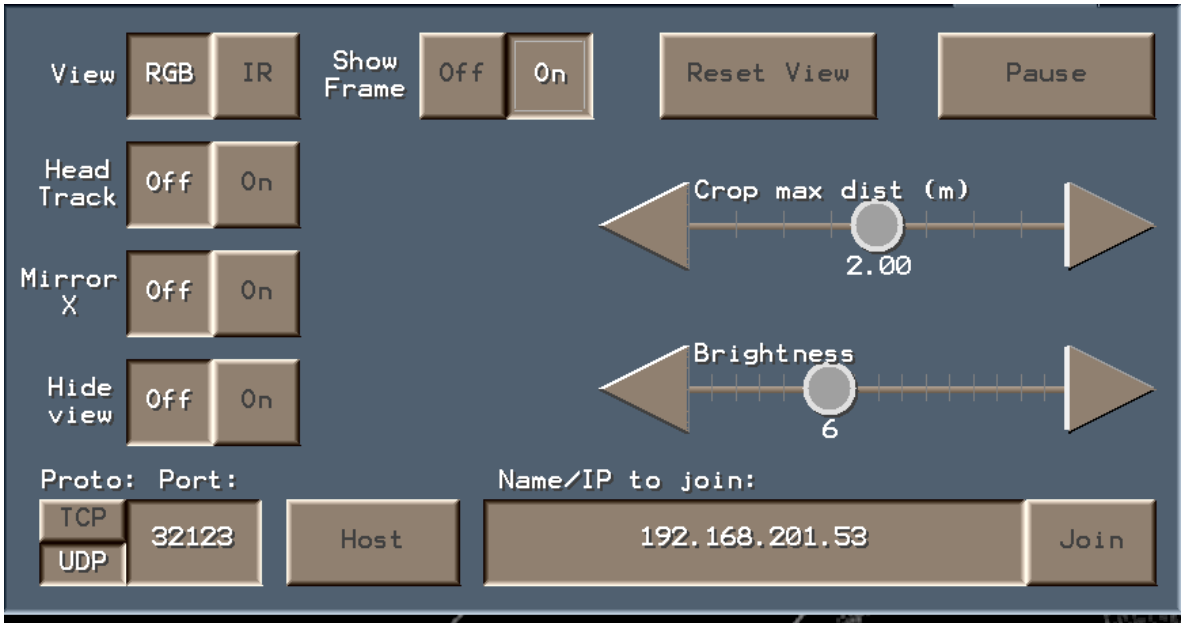

Figure 24 KniView's VoxieMenu tab.

There isn't much that the *KniView.ini* offers that can't be done and adjusted in the in app menu.

A description of each option is as follows

Option	**Description**
RGB / IR	Switch between the RGB and Infrared view
Show Frame	If enabled a wireframe is drawn around the outside of the volume
Reset View	Resets the current view to default values
Pause	Pauses the playback of the footage
Head Track	Attempts to find the 'head' of a person in the image and centres places the image in the centre of the volume.
Crop Max Dist (m)	Crops out the max distance. To trim the depth of the image.
Mirror X	Mirrors the output of the footage
Hide view	If enabled the captured footage will be hidden

Brightness	Brightness setting for the capture footage. Lower value less voxels and darker Higher value more voxels and brighter
Proto TCP / UDP	selecting networking protocol to use
Host -	Enable this button to host the network session
Port	For networking the port number to attempt to connect through
Name/IP to join	For networking the IP address to attempt to join (the host's IP address)
Join	Press to join a hosted networking session using the nominated port and IP address

KniView.ini contents and options

KniView stores various settings within *KniView.ini*. Most of these settings can be adjusted in the app, but there are a few networking settings which can be used to optimise the connection.

Note: *KniView.ini* also stores the pos&ori values for its media files. These are amended onto the end of the file.

Option (and default value)	**Description**
Input=k	Default input type (o=OpenNI2 (Kinect 1.0), k=Kinect 2.0, r=RealSense 2.0 (SR300/D415) a=Azure)
output=w;	Selects output mode (w= 2D window, 3= 3DVison, v=Volumetric (VX app))
infrared=0;	Launch in RGB or infrared display mode (0 for RGB and 1 for Infrared)
brightness=4;	Brightness setting for footage value between (0 … 10). Higher value brighter
showframe=1;	If enabled a wireframe border is presented around the display.
proto=1;	Select which network protocol to use 0=TCP, 1=UDP
port=32123	Set which port to connect to
join=192.168.201.53	Set Host's IP address to join
limitfps=15;	Limit the FPS to be sent and received through a network connection //use 1.0 for WIFI; 15.0 for fast connection
muteit=0;	If enabled play no sound
udp_bylimit=1500;	UDP connection byte limitation
udp_boxskip=1;	UDP connection setting
depmax=2.0;	Capture's depth max value

qualcol=15;	jpeg quality for RGB (1 (fewest bytes) .. 100 (most bytes)) for compression over network
qualdep=99;	depth quality (1 (fewest bytes .. 100 (most bytes)) for compression over network
qualinf=50;	jpeg quality for infrared (1 (fewest bytes) .. 100 (most bytes)) for compression over network

Using Knife

Unlike *KniView*, *Knife* can only be a Windows app. It can record, edit and playback recorded .kni media. ***Knife* is the only way to record depth camera footage.**

Similar to *KniView*, *Knife* is a command line application. Running *Knife* with no command line parameters will assume it is in live capture mode. To load a .kni file for playback it needs to be specified in the command line. For example:

KNIFE.EXE command line:

knife <-- run live video from Kinect v2/Senz3D

knife rec0000.kni <-- play looped video from specified file

The *Knife.txt* file within the *KniView* folder has more detailed instructions on how to use the *Knife* application.

Using Knife to record a new KNI video

To record a new .kni video. Run *Knife* in live capture mode (without any extra parameters) and once you are happy with the setup, press the record button on the screen. The new Kni file will be written to the same directory that *knife.exe* is in and will be labeled as *REC###.KNI* and number sequentially from low to high.

Using Knife to playback an existing KNI video

To playback an existing .kni video, use the command prompt to type out the .kni file you wish to load. You can use relative or full paths to load in your file.

Knife C:\Voxon\Videos\MyCapture.kni

Knife MyCapture.kni

Knife \Media\MyCapture.kni

Are all valid ways to launch a .kni file (assuming the filename exists).

Knife GUI options

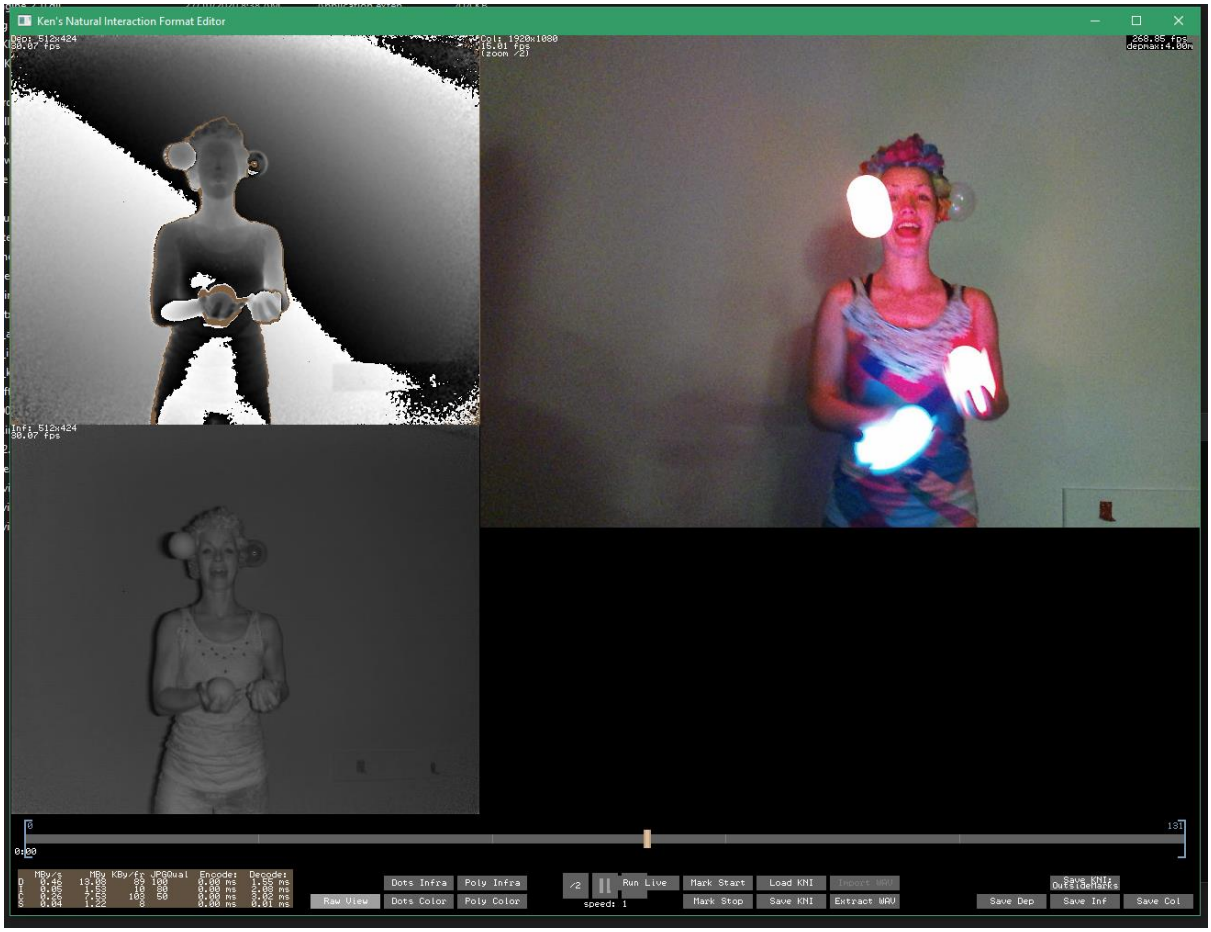

Figure 25 Knife's GUI

Knife's main display shows the view type. Raw type shows the depth, infrared and color layers. Other view types show a dot based or poly view of infrared or color views.

Underneath the display is the timeline. This shows the starting and ending marker, the current frame, the length in seconds of the recorded footage, and the frame number.

The bottom right shows various stats about the filetype, how much data is being captured by each data layer (D = depth, I = infrared, C = color, S = sound) and both the encode and decode speeds.

A description of the the various buttons at the bottom of the screen are as follows

Button	**Description**
Raw View	Switch to Raw view of the 3 layers (depth, infrared and colour)
Dots Infra	Show the Infrared image in a dot view
Dots Color	Show the colour image in a dot view
Poly Infra	Show the Infrared image as a poly / mesh view
Poly Color	Show the colour image as a poly / mesh view
Mute	Mute / Unmute the sound
Speed /2	Playback footage at half speed
Pause (II button)	Pause the playback of footage
Speed *2	Playback footage at double speed
Run Live	Switch to live capture
Mark Start	Set start marker to where the current position is
Mark Stop	Set stop marker to where the current position is
Load KNI	Load an existing KNI recording
Save KNI	Save current KNI file trimed to be between the start and stop markers
Import Wav	Import audio wav file to replace current audio
Export Wav	Export audio file as a wav file from Kni
Save KNI; Outside Marks	Save the whole KNI file ignoring the start and stop markers.
Save Dep	Export the depth image as a PNG file
Save Inf	Export the Infrared image as JPG file
Save Col	Export the colour image as a JPG file

Using Knife to edit an existing KNI video

Any KNI that has been loaded though Knife can be edited.

Editing a .kni file is limited. A .kni file can be top and tailed. That is the start and end points can be adjusted. This is done by moving the two blue '[' and ']' bracket markers on each side of the timeline on the *Knife* GUI and then choosing the 'Save KNI'. Choosing 'Save KNI: outside marks' will save the whole capture and discard the start and ending markers.

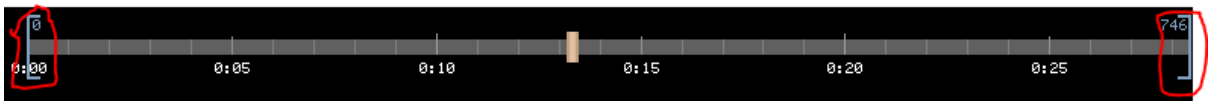

Figure 26 Knife's timeline revealing the blue markers.

It is important to note a more advanced editing suite for KNI is possible. If you are interested in exploring this use case, please contact Voxon Photonics directly.

Keyboard controls for Knife

Keyboard Command	Description
ESC	quit
1-5	set render mode: 1:2D, 2:IR dots, 3:RGB dots, 4:IR poly, 5:RGB poly
Space Bar or P	pause (/play if playing KNI)
M	Toggle mute
R	record video & audio to next REC####.KNI file (start/stop)
/	reset pos & ori for 3D view
Arrows/RCtrl/Numpad 0	move 3D camera
, or.	rotate view left/right around center
L.Shift	fly 16x faster
R.Shift	fly 16x slower
- or =	adjust max depth (3D view) in steps of 250 (500..8000)
Q/W/E	toggle head track mode (Q=off, W=show circle, E=full debug)
F12	screen capture to: depth:KIND####.PNG, IR:KINI####.PNG, RGB:KINC####.PNG

Replacing the audio of an existing KNI video

Audio files within the .kni file are uncompressed 44.1 KPS 16 bit .wav files.

Knife can export out the embedded audio file within a .kni file, and a new wav file can be imported into a .kni file.

Importing a new audio file can be done through *Knife* or via the standalone command line application. *kninew_wav.exe* which is located in the *KniView* directory. Syntax is as follows:

"Kni_newwav KniFileToEdit.kni wavefiletoEncode.wav 0 (the offset in seconds to encode the wave)"

The output will always be called *temp.kni* so make sure you rename your new .kni file appropriately.