PVRGeoPOD

User Manual

Filename	:	PVRGeoPOD.User Manual.1.15f.External.doc
Version	:	1.15f External Issue (Package: POWERVR SDK 2.05.25.0755)
Issue Date	:	07 Jul 2009
Author	:	POWERVR

Contents

List of Figures

1. Introduction

PVRGeoPOD is a file-exporter plug-in for 3D Studio MAX and Maya. Some notable supported features are as follows:

- Parented nodes
- Mesh instancing
- Can export as many texture-coordinate sets as MAX and Maya supports
- Boned meshes
- Bone batching (can split a mesh into sub-meshes when the mesh has more bones than can fit into the bone palette)
- Complete choice of data formats (e.g. floats, bytes, …)
- Choice of interleaved vertex data or separate channels (e.g. position, normal, texture coordinates).
- Tangent space generation
- Polygon/vertex sorting
- Polygon stripping

1. Installation

1.1. 3ds max

1.1.1. 3ds Max Data Exchange Interface (3DXI) - previously known as IGame

PVRGeoPOD for 3ds max uses Autodesk's 3DXI interface, which relies on a file called "igame.dll" which resides in the 3ds max installation directory. This file is frequently updated by Service Packs for 3ds max, as and when Autodesk (and previously Discreet) releases them, and by specific updates they may release for 3DXI (previously IGame).

Unfortunately, the various versions of "igame.dll" are often incompatible, and only one may be installed at any point in time. It is therefore very difficult to use two different 3ds max plug-ins that both use 3DXI, but are compiled to use different versions of it.

PVRGeoPOD attempts to work around this issue by having each build of the plug-in simultaneously support as many versions of 3ds max and 3DXI as is possible and practical.

1.1.2. Supported configurations

Plug-in	3ds max release	"igame.dll" version
`3dsmax6\PVRGeoPOD.dle`	`3ds max 6`	`6.0.0.56`
	`3ds max 6 + SP1`	`6.0.1.62`
	`3ds max 7`	`7.0.0.65`
	`3ds max 7 + SP1`	`7.0.1.76`
	`3ds max 7 + 3DXI 2.0`	`7.0.1.78`
	`3ds max 8 + 3DXI 2.0`	`8.0.0.40`

	`3ds max 8`
	`(contains winding order bug?)`
		`8.0.0.92`

	`3ds max 8 + SP1`	`8.0.1.11`
	`3ds max 8 + SP2`	`8.0.1.18`
	`3ds max 8 + SP3`	`8.0.1.24`
`3dsmax9\PVRGeoPOD.dle`	`3ds max 9`	`9.0.0.100`
	`3ds Max 2008`	`10.0.0.86`
	`3ds Max 2009`	`11.0.0.57`
	`3ds Max 2010`	`12.0.0.106`

1.1.3. Installation

3ds max 6

Copy "`3dsmax6\PVRGeoPOD.dle`" to "`C:\3dsmax6\plugins\`", or the equivalent location for where your copy of 3ds max is installed.

3ds max 7

Copy "`3dsmax6\PVRGeoPOD.dle`" to "`C:\3dsmax7\plugins\`", or the equivalent location for where your copy of 3ds max is installed.

3ds max 8

Copy “`3dsmax6\PVRGeoPOD.dle`” to “`C:\Program Files\Autodesk\3dsMax8\plugins\`”, or the equivalent location for where your copy of 3ds max is installed.

3ds max 9

Copy “`3dsmax9\PVRGeoPOD.dle`” to “`C:\Program Files\Autodesk\3ds Max 9\plugins\`”, or the equivalent location for where your copy of 3ds max is installed.

3ds Max 2008

Copy “`3dsmax9\PVRGeoPOD.dle`” to “`C:\Program Files\Autodesk\3ds Max 2008\plugins\`”, or the equivalent location for where your copy of 3ds max is installed.

3ds Max 2009

Copy “`3dsmax9\PVRGeoPOD.dle`” to “`C:\Program Files\Autodesk\3ds Max 2009\plugins\`”, or the equivalent location for where your copy of 3ds max is installed.

3ds Max 2010

Copy “`3dsmax9\PVRGeoPOD.dle`” to “`C:\Program Files\Autodesk\3ds Max 2010\plugins\`”, or the equivalent location for where your copy of 3ds max is installed.

1.2. Maya

1.2.1. Installation

Maya 7

Windows

Copy “`PVRGeoPOD_v7.mll`” to “`C:\Program Files\Alias\Maya7.0\bin\plug-ins\`”, or the equivalent location for where your copy of Maya is installed.

Linux

Copy “`PVRGeoPOD_v7.so`” to “`/usr/aw/maya7.0/bin/plug-ins/`”, or the equivalent location for where your copy of Maya is installed.

Once the plug-in is copied to the correct location enable it in Maya’s plug-in manager.

Maya 8

Windows

Copy “`PVRGeoPOD_v8.mll`” to “`C:\Program Files\Alias\Maya8.0\bin\plug-ins\`”, or the equivalent location for where your copy of Maya is installed.

Linux

Copy “`PVRGeoPOD_v8.so`” to “`/usr/aw/maya8.0/bin/plug-ins/`”, or the equivalent location for where your copy of Maya is installed.

Once the plug-in is copied to the correct location enable it in Maya’s plug-in manager.

Maya 8.5

Windows

Copy “`PVRGeoPOD_v8.5.mll`” to “`C:\Program Files\Autodesk\Maya8.5\bin\plug-ins\`”, or the equivalent location for where your copy of Maya is installed.

Linux

Copy “`PVRGeoPOD_v8.5.so`” to “`/usr/aw/maya8.5/bin/plug-ins/`”, or the equivalent location for where your copy of Maya is installed.

Once the plug-in is copied to the correct location enable it in Maya’s plug-in manager.

Maya 2008

Windows

Copy “`PVRGeoPOD_v2008.mll`” to “`C:\Program Files\Autodesk\Maya2008\bin\plug-ins\`”, or the equivalent location for where your copy of Maya is installed.

Linux

Copy “`PVRGeoPOD_v2008.so`” to “`/usr/autodesk/maya2008/bin/plug-ins/`”, or the equivalent location for where your copy of Maya is installed.

Once the plug-in is copied to the correct location enable it in Maya’s plug-in manager.

Maya 2009

Windows

Copy “`PVRGeoPOD_v2009.mll`” to “`C:\Program Files\Autodesk\Maya2009\bin\plug-ins\`”, or the equivalent location for where your copy of Maya is installed.

Once the plug-in is copied to the correct location enable it in Maya’s plug-in manager.

2. Supported File Formats

2.1. POD File

PVRGeoPOD can also export to a binary format (file extension: “POD”). The POWERVR SDK Tools library, part of the POWERVR SDK, contains code to load and save POD files.

```
CPVRTModelPOD m_model;

// Load the model
if(!m_model.ReadFromFile(“model.pod”))
        return false;

// Do stuff

// Free the memory
m_model.Destroy();
```

The advantages of a POD file:

1. A POD file can be loaded into memory, the vertex data copied into HW friendly buffers – e.g. a vertex buffer object (OpenGL [ES]) or a vertex buffer (Direct3D [Mobile]) - and the POD file released, or even just the mesh-data memory freed (in which case set the freed memory pointers to NULL).
   This saves memory, compared to headers, on operating systems which do not support page files: the memory used by the data in a header file will be around for the lifetime of the application, and therefore two copies of mesh data are in memory. This is not a problem for operating systems which support a page-file, since the unused memory can be swapped out. On OpenGL [ES] this is not an advantage if `glVertexPointer()` and its ilk are used, since no copy is required.
2. It also allows POD files to be changed without rebuilding the application.
3. It is smaller on disk (although there should be little size difference between two compiled executables, one using a header file and one with a POD file in resources).

2.1.1. Example

See *ChameleonMan* in the POWERVR SDK.

2.2. Header and Source File

PVRGeoPOD can export the binary POD file directly to a header-file (file extension: “H”) or a cpp-file (file extension: “CPP”) as if it had been wrapped using the POWERVR SDK’s FileWrap utility.

To load from a cpp-file follow the instructions for loading a conventional POD binary file. To load from a header-file use the POWERVR SDK Tools library ReadFromMemory function.

```
// Include the scene data
#include “model.h”

CPVRTModelPOD m_model;

// Load the model
if(!m_model.ReadFromMemory(_model_pod, _model_pod_size))
        return false;

// Do stuff

// Free the memory
m_model.Destroy();
```

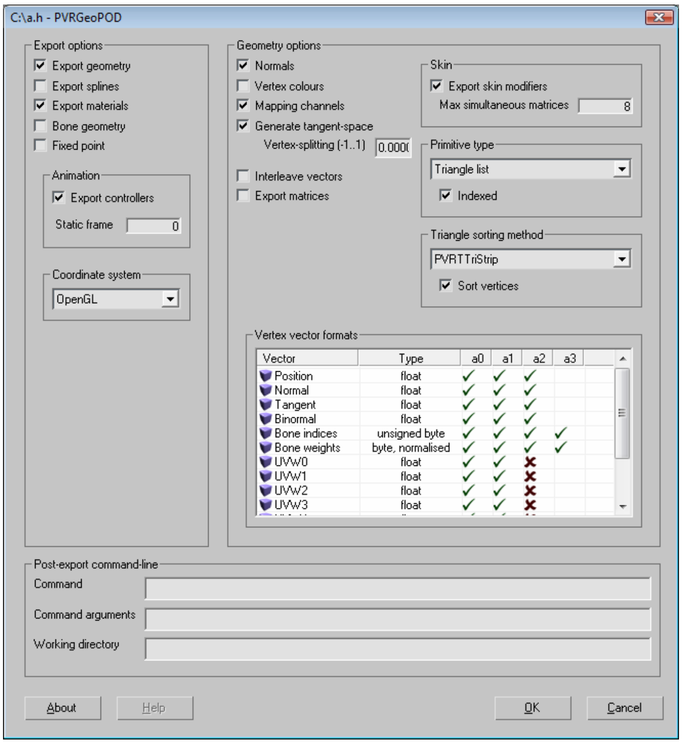
3. Using the Exporter

3.1. 3D Studio Max

PVRGeoPOD can be used in 3ds max by:

1. From the menu, select “`File/Export…`” or “`File/Export Selected…`”.
2. In the “`Save as type`” drop-down box, select “`POWERVR Exporter (*.POD,*.H,*.cpp)`”.
3. Browse to the location you wish the file to be saved.
4. Double-click an old file to overwrite, or type in a filename; use the extension .H for a header-file, and .POD for a binary file. If the filename does not specify an extension, POD will be assumed.

You will now see the PVRGeoPOD dialog-box, as shown in Figure 1.



C:\a.h - PVRGeoPOD

Export options
- ☑ Export geometry
- ☐ Export splines
- ☑ Export materials
- ☐ Bone geometry
- ☐ Fixed point

Animation
- ☑ Export controllers
- Static frame: 0

Coordinate system: OpenGL

Geometry options
- ☑ Normals
- ☐ Vertex colours
- ☑ Mapping channels
- ☑ Generate tangent-space
- Vertex-splitting (-1..1): 0.000(
- ☐ Interleave vectors
- ☐ Export matrices

Skin
- ☑ Export skin modifiers
- Max simultaneous matrices: 8

Primitive type: Triangle list
- ☑ Indexed

Triangle sorting method: PVRTTriStrip
- ☑ Sort vertices

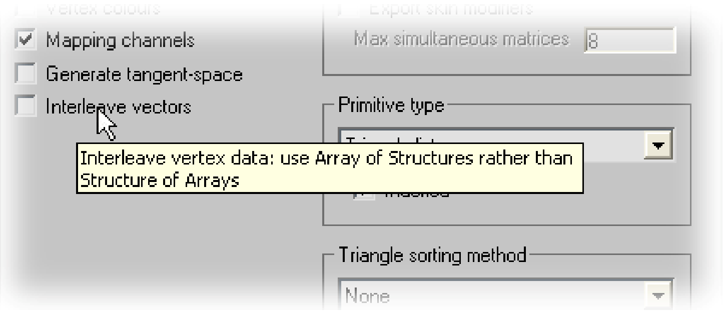
Vertex vector formats

Vector	Type	a0	a1	a2	a3
Position	float	✓	✓	✓	
Normal	float	✓	✓	✓	
Tangent	float	✓	✓	✓	
Binormal	float	✓	✓	✓	
Bone indices	unsigned byte	✓	✓	✓	✓
Bone weights	byte, normalised	✓	✓	✓	✓
UVW0	float	✓	✓	✗	
UVW1	float	✓	✓	✗	
UVW2	float	✓	✓	✗	
UVW3	float	✓	✓	✗	

Post-export command-line
- Command:
- Command arguments:
- Working directory:

Figure 1 The PVRGeoPOD dialog box

The function of each dialog-box element is individually described in a tool-tip that will appear when the mouse is hovered over them, as shown in Figure 2.



- ☑ Mapping channels
- ☐ Generate tangent-space
- ☐ Interleave vectors

Max simultaneous matrices: 8

Primitive type

Interleave vertex data: use Array of Structures rather than Structure of Arrays

Triangle sorting method: None

Figure 2 Everything has a tool-tip

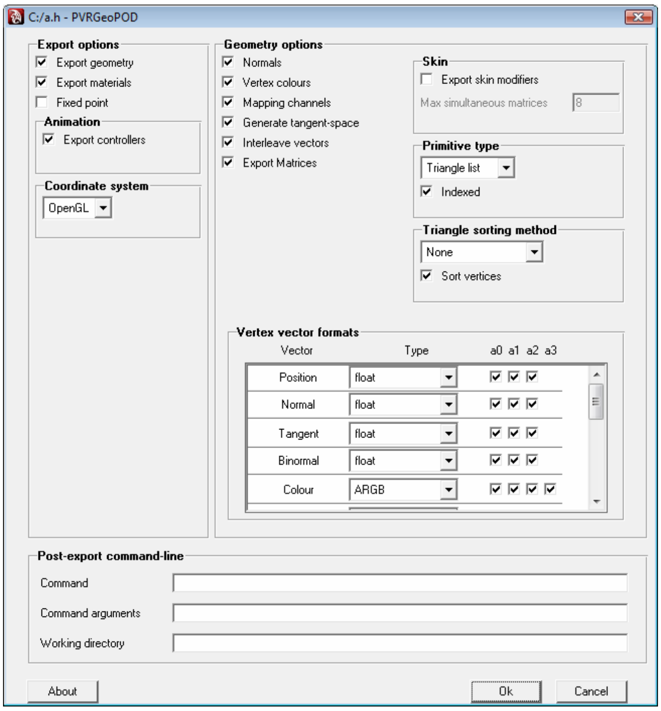
More detailed help is available from the “`Help`” button at the bottom of the window.

3.2. Maya

PVRGeoPOD can be used in Maya by:

5. From the menu, select “`File/Export All`…” or “`File/Export Selection`…”.
6. In the “`Files of type:`” drop-down box, select “`POWERVR Exporter (*.pod,*.h and *.cpp)`”.
7. Browse to the location you wish the file to be saved.
8. Double-click an old file to overwrite, or type in a filename; use the extension .h for a header-file, and .pod for a binary file. If the filename does not specify an extension then the file will be saved in POD format.

You will now see the PVRGeoPOD dialog-box, as shown in Figure 3.



C:/a.h - PVRGeoPOD

Export options
- ☑ Export geometry
- ☑ Export materials
- □ Fixed point

Animation
- ☑ Export controllers

Coordinate system: OpenGL

Geometry options
- ☑ Normals
- ☑ Vertex colours
- ☑ Mapping channels
- ☑ Generate tangent-space
- ☑ Interleave vectors
- ☑ Export Matrices

Skin
- □ Export skin modifiers
- Max simultaneous matrices: 8

Primitive type: Triangle list
- ☑ Indexed

Triangle sorting method: None
- ☑ Sort vertices

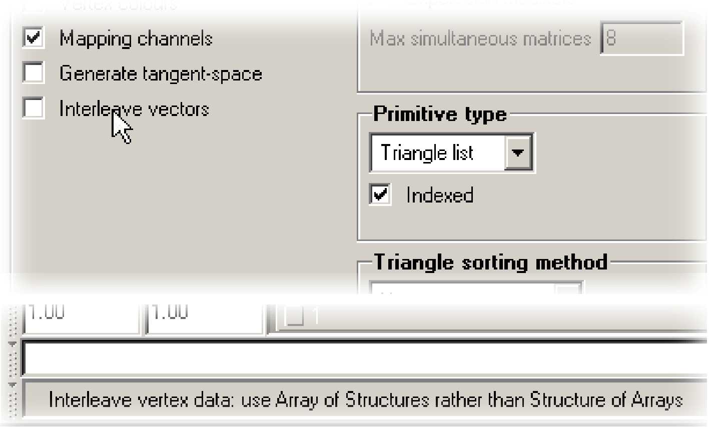
Vertex vector formats

Vector	Type	a0	a1	a2	a3
Position	float	☑	☑	☑	
Normal	float	☑	☑	☑	
Tangent	float	☑	☑	☑	
Binormal	float	☑	☑	☑	
Colour	ARGB	☑	☑	☑	☑

Post-export command-line
- Command:
- Command arguments:
- Working directory:

Figure 3 The PVRGeoPOD dialog box (Windows version)

The function of each dialog-box element is individually described in an annotation that will appear in Maya’s Help Line when the mouse is hovered over them, as shown in Figure 4.



- ☑ Mapping channels
- □ Generate tangent-space
- □ Interleave vectors

Max simultaneous matrices: 8

Primitive type: Triangle list
- ☑ Indexed

Triangle sorting method

Interleave vertex data: use Array of Structures rather than Structure of Arrays

Figure 4 Everything has an annotation

4. Using POD Data in your Application

The POWERVR Tools library, part of the POWERVR SDK, contains code to load and save POD files; it also contains many utility functions to help use the resultant data structures, which can be used both when the data is loaded from a POD file **and** when compiled in from a header file.

The POD data structures are relatively straightforward and easy to use. For example, in OpenGL ES, the vertex (position) data pointer can be used directly as an input to `glVertexPointer()`; the same goes for the pointer to the normals, and all the others.

For tasks such as acquiring World matrices, bone matrices, etc, you can use the code in the `PVRTModelPOD` module from the POWERVR Tools library.

Appendix A. POD File Format

This information is irrelevant if the code supplied in “`PVRTModelPOD.cpp`” - to load binary POD files - is used, as we recommend.

A.1. Binary File (“POD”)

A.1.1. Binary File Format

POD files are stored in a tagged, nested structure. A marker consists of two 32-bit values:

DWORD	Bit	Symbol	Description
0	31	End	End of data block bit 0 This marker is the beginning of a block 1 This marker is at the end of a block
	30:0	Name	Marker name, identifying the data which follows
1	31:0	Length	Amount of data which follows; AKA distance to next marker.

- After each marker, `Length` bytes of data follow.
- The list of possible `Name` values is in the enumerated type `EPODFileName` in the file “`PVRTModelPOD.cpp`”.

A.1.2. Overview of File Reading

```
10 Read marker
20 If recognised marker, read the following data, otherwise skip it
30 GOTO 10
```