

Ninja PVRConv

(SGI/WINDOWS 版)

1998/09/04

VERSION : 0.85alpha1

09/04/98 0.85alpha1

- ・ Twiddled Rectangle 形式に対応。
- ・ VQ4 対応。
- ・ repeat を指定しない時のミップマップ生成の不具合を修正。

08/07/98 0.84alpha1

- ・ B M P ファイルからのパレット P V R , P V P に対応。 ARGB8888, RGB565, ARGB4444, ARGB1555 形式のパレット出力が可能。

07/10/98 0.82alpha1

- ・ Y U V テクスチャ、バンプマップテクスチャの出力ができます。
- ・ -repeat オプション追加。今までの pvrconv ではミップマップ生成時に無条件に横方向のピクセル間での補完をしていたため両端に反対側のエッジの色が混じるという現象がありました。これを改善するため両端のピクセル間の補完をしないアルゴリズムに切り替えています。-repeat オプションで従来までのリピートテクスチャミップマップが生成できます。
- ・ SET5 以降性能を出すためにはアライメント調整が必須となります。今回.pvr ファイルのデータの先頭を 32 バイトアライメント調整しました。具体的には GlobalIndex チャンクサイズを 12 バイトから 16 バイトにします。従来のものに後ろ 4 バイトのダミーデータをつけています。

[GBIX][4][globalIndex] [GBIX][8][globalIndex][0]

今後もテクスチャ回りでのアライメント調整が入る可能性があります。

05/31/98 0.78alpha1

- ・ ディザ、アルファディザをデフォルトで off するようにしました。実際ディザを off しているユーザが多いこと。vq への対応によりデフォルトでディザ off の方がわかりやすいためです。
- ・ VQ に対応しました。アルゴリズムのバージョンとして vq3 と呼びます。現在 vq4 を準備中です。
- ・ smallVQ に対応しました。smallVQ とは一つのコードブックに複数のテクスチャを格納することで小さなテクスチャ 32×32、16×16 で VQ 圧縮をするものです。使用に関しては smallVQ 対応されたライブラリと組み合わせる必要があります。
- ・ テクスチャを上下逆さまにする flip オプションをサポート。
- ・ -outname で出力ファイル名を指定できます。

03/23/98 0.73alpha1

- ・ 変更なし

02/04/98 0.72alpha1

- ・ テクスチャコンバートのリバーズ 対応。オプションで 値を 255- にして変換。

12/10/97 0.7alpha3

- ・ -ns オプションを追加。テクスチャの suffix を無視する。ファイル名から拡張子をはずした名前で変換をする。
- ・ テクスチャサイズが 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 の場合拡張子がないテクスチャでも変換できるようにした。pic, tga, bmp, pix を検出して自動で変換。

11/07/97 0.7alpha2

- ・ 二つの bmp からアルファ付きの pvr を作る部分を実装していなかったのをこれを作成。
- ・ 512x512, 1024x1024 の tga が変換できなかったのをこれを改修。
- ・ 512x512, 1024x1024 の twiddled の変換で左上 1 / 4 以外が黒くなり正しく変換できないバグを改修。
- ・ mipmap なしの場合に globalIndex チャンクが pvr ファイルに書き込まれないバグを改修。

PVRConv の使用方法

PVRConv は次のフォーマットのテクスチャを PowerVR 用のテクスチャ.pvr ファイルに変換します。

.pic : softimage テクスチャ
.bmp : P C ビットマップファイル
.tga : Light Wave 3D, 3D Studio MAX
.pix, .mask : Alias テクスチャファイル。 .mask は イメージファイル。

pvr ファイルは I F F チャンク形式を利用しています。Ninja はプロジェクトを通してのテクスチャ通し番号である globalIndex を使用します。PVRConv は globalIndex チャンク (チャンネル名 GBIX) と PowerVR 用テクスチャチャンク (チャンネル名 PVRT) の二つのチャンクを pvr ファイルに書き込みます。

Ninja ライブラリで使う場合、必ず globalIndex を指定してください。PVRConv のコンバータコア部分は各ツールの NinjaExport の中から呼び出されています。

Usage:

Version 0.85alpha1

-<command line>-

convert : pvrconv [options] <infile1> [<outpath>]
convert : pvrconv [options] <infile1.bmp> [<alpha.bmp>] [<outpath>]
convert : pvrconv [options] <infile1.pix> [<alpha.mask>] [<outpath>]
infile: .bmp .pic .tga .pix(.mask)
outfile: .pvr

.bmp を入力する場合

値指定がない場合は infile1 として bmp ファイルを指定します。 がある場合、bmp は を持つことができないため二つの bmp ファイルを使って表現します。

PVRConv -gi 100 rgb.bmp alpha.bmp

ここで alpha.bmp の R 値が 値として利用されます。

.pix, .mask を入力する場合

Alias では.pix ファイルをテクスチャとして、.mask ファイルを として使います。したがって、Alias において 付きのテクスチャを pvr ファイルにコンバートする場合は、.pix, .mask の両方を引数とします。

PVRConv -gi 101 texture.pix alpha.mask

Alpha を使わない場合は.pix ファイルのみでよい。

次にコンバータのデフォルトアクションを示します。 値、正方形、長方形をチェックし自動的に出力を切り替えます。

<converter default action>

square texture : twiddled
rectangle texture : rectangle
auto mipmap : ON
dither : OFF
alpha dither : OFF
stride : OFF
pixel format : 565(no translucent)
 : 4444(translucent alpha0-255)
 : 1555(bilevel translucent alpha0,255)

<VQ>

VQ auto mipmap : ON

PVRConv は、もとの画像の 値をチェックし、自動的に次の三つのフォーマットを切り替えて出力する。

がない場合 : RGB565 で出力。
がある場合 : ARGB4444 で出力。
が 0 , 2 5 5 の二値の場合 : ARGB1555 で出力。

またテクスチャが正方形の場合は twiddled 形式が、長方形の場合は rectangle 形式がコンバータで自動選択される。twiddled の場合ミップマップも自動生成。

twiddled 形式

ピクセルをメモリから高速で読み出すことができる順番に並べ替えたテクスチャ。Mipmap が利用可能。表示が高速。

rectangle 形式

ピクセルの順番をイメージそのままとしているテクスチャ。表示が twiddled に比べ低速。
Mipmap が使えないので注意。

VQ は mipmap を使うと画質が落ちるためユーザ選択として VQ 以外は mipmap ありで VQ は mipmap なしができるように VQ の mipmap オプションを分けている。auto モードでは VQ 以外、VQ とともに mipmap が ON になっている。

-<options>-

-t or -twiddled : twiddled texture

正方形のテクスチャの場合、自動的に twiddled になるため指定の必要はありません。長方形テクスチャに指定した場合、Twiddled rectangle として出力する。

-r or -rectangle : rectangle texture

正方形テクスチャに対し指定した場合、twiddled にせずに rectangle として出力できる。

-s or -stride : stride texture

stride テクスチャとして出力。Stride テクスチャは rectangle テクスチャの一部の領域を 3 2 ピクセル単位のサイズで横幅指定できる。Stride テクスチャは Stride レジスタに値を書き込むことにより幅を指定するため一つのみ。

-bu or -bump : bump texture

bump テクスチャとして出力。ピクセル値の濃淡を bump 法線値に変換して bump テクスチャを生成します。

-v3 or -vq3 : VQ algorithm 3

VQ 3 テクスチャとして出力。32×32 以下は VQ 3 にすると逆にデータ量が大きくなるため 64×64 以上を対象に VQ 3 をする。正方形でないテクスチャは rectangle に変換。

-v3a or -vq3auto : VQ and smallVQ(32x32, 16x16) algorithm 3

32×32 以下のテクスチャを自動的に smallVQ 3 に 64×64 以上を VQ に変換する。

-sv3 or -smallvq3 : smallVQ algorithm 3

smallVQ 3 テクスチャとして出力。SmallVQ 3 がサポートするサイズは 64×64、32×32、32×32MM、16×16、16×16MM の五種類のみ。これはコードブックをシェアすることでデータ圧縮率得られるもののみの対応のためである。

-v4 or -vq4 : VQ algorithm 4

VQ 4 テクスチャとして出力。32×32 以下は VQ 4 にすると逆にデータ量が大きくなるため 64×64 以上を対象に VQ 4 をする。正方形でないテクスチャは rectangle に変換。

-v4a or -vq4auto : VQ and smallVQ(32x32,16x16) algorithm 4

32×32 以下のテクスチャを自動的に smallVQ 4 に 64×64 以上を VQ 4 に変換する。

-sv4 or -svq4 : smallVQ algorithm 4

smallVQ 4 テクスチャとして出力。SmallVQ 4 がサポートするサイズは 64×64、32×32、32×32MM、16×16、16×16MM の五種類のみ。これはコードブックをシェアすることでデータ圧縮率得られるもののみの対応のためである。

-v4p or -vq4pri : use priority data for VQ algorithm 4

VQ4 アルゴリズムにプライオリティデータを使用します。前述の 値設定用画像ファイルがある場合には 3 番目のファイル名として、そうでない場合は、2 番目のファイル名として、プライオリティデータに使用する BMP 画像を指定します。VQ 4 アルゴリズムは、指定された BMP 画像の R 値をプライオリティデータとして使用します。R 値が大きければプライオリティも高くなり、その部分の品質をなるべく保持しようとしています。

(例) プライオリティデータを用いた VQ 4 テクスチャを生成する場合

(指定ファイルがある場合)

PVRConv -gi 102 -vq4 -vq4pri test.bmp alpha.bmp priority.bmp

(指定ファイルがない場合)

PVRConv -gi 102 -vq4 -vq4pri test.bmp priority.bmp

-p8 or -palette8 : palette format Source Image 8bpp 256 color

-p4 or -palette4 : palette format Source Image 4bpp 16 color

パレットタイプテクスチャとして PVR ファイルを出力する。パレットデータは PVP ファイルに格納される。また、変換可能なソースイメージは BMP のみである。ソースの BMP のパレット数は、8 bpp ならば 256 色分、4 bpp ならば 16 色分付加せねばならない。

また、アルファ用の BMP を使用することも可能です。この場合、パレットデータの RGB 値の内、R 値をアルファとして使用する。アルファ用 BMP が無い場合は、0xFF がアルファ

値として設定される。あくまでパレット値としてのアルファ値が変わるだけです。

-5 or -565 : pixel format RGB565

テクスチャフォーマット RGB565 を指定する。

-4 or -4444 : pixel format ARGB4444

テクスチャフォーマット ARGB4444 を指定する。

-1 or -1555 : pixel format ARGB1555

テクスチャフォーマット ARGB1555 を指定する。

テクスチャフォーマット ARGB8888 を指定する。

-8 or -8888 : pixel format ARGB8888

テクスチャフォーマット ARGB8888 はパレットタイプテクスチャのみです。

-y or -yuv : pixel format YUV422

テクスチャフォーマット YUV422 を指定する。YUV422 は隣り合う 2 ピクセル間で YUV のうちの UV 値に同じ値を使うこと ((Y0,U), (Y1,V) を 8 ビット精度でイメージに格納、ここで $U = (U0+U1)/2$ 、 $V = (V0+V1)/2$ で YUV 各 8 ビットの階調を確保。ハードウェアは YUV を描画時に RGB 値に戻して ((Y0,U,V), (Y1,U,V) から 2 ピクセルを RGB に変換) フレームバッファに書き込みます。RGB のテクスチャに比べ階調がきれいだが横方向に同じ UV 値を使うため横方向の解像度が落ち斜線などはつぶれます。

-t1 or -t1555 : translucent pixel format ARGB1555

2 値でない半透明の場合 ARGB4444 に変換されるが、これを ARGB1555 に変換する。

-b4 or -b4444 : bilevel translucent pixel format ARGB4444

2 値の半透明の場合 ARGB1555 に変換されるが、これを ARGB4444 に変換する。

-ra or -ralpha : reverse alpha (0 -> FF, FF -> 0)

アルファ値 a を 255-a に置き換えて変換する。抜きたい部分を逆にしてしまった場合に使用

する。

-di or -dither : dither on

ディザを on する。

-adi or -adither : alpha dither on

アルファディザを on する。

-ns or -nosuffix : ignore suffix

入力されたテクスチャの拡張子を無視したファイル名を指定する。外部ツールでテクスチャのフォーマットを変更した場合でも、モデルが持つテクスチャ名を変更なしで変換できる。この場合テクスチャ名は拡張子なしにする必要があります。テクスチャサイズが 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 の場合拡張子がなくても pic, tga, bmp, pix を検出して自動で変換。

-f or -flip : テクスチャを上下反転する。

ミップマップ自動生成を off する。

-nm or -nomipmap : auto mipmap off

ミップマップ自動生成を off する。

-nvm or -novqmipmap : VQ auto mipmap off

VQ コンバート時のミップマップ自動生成を off する。

-rep or -repeat : Repeat on

ミップマップ生成時にエッジのつながりを考慮する。

-ps or -palsetting <bank no>,<start palette>,<palette number> : palette information setting

パレットデータセーブ時に必要な情報、バンクナンバー、バンク内の開始パレット番号、登録パレット数を設定する。格納するパレットの位置、数を設定できる。デフォルトの設定値は、

8 bpp	: バンク 0 番	開始パレット 0 番	格納パレット数 2 5 6 個
4 bpp	: バンク 0 番	開始パレット 0 番	格納パレット数 1 6 個

設定例：

-ps 3,53,102 バンク 3 番 開始パレット 53 番 格納パレット数 1 0 2 個

-p or -path <dir> : output path

出力ディレクトリを指定する。このオプションを使わなくてもファイル名の後ろにディレクトリ名を書けば同様の動作をする。

PVRConv -gi 102 test.tga /tmp

PVRConv -gi 102 test.tga -p /tmp

-gi or -globalIndex: set globalIndex chunk

GBIX チャンクに指定された globalIndex を書き込みます。

-out or -outname <filename> : output filename

出力ファイル名を指定する。

以 上