

計算リソースはCPUだけじゃない!  
画像処理の高速化や演算処理のオフロードに

# ラズベリー・パイで始める GPUプログラミング

## 第3回(最終回) GPGPUプログラミングにトライ

本橋 弘臣

リスト1 画像白黒反転プログラム(shader-invert.hの一部抜粋)

前回使用したshader-copy.hをベースに1行修正する

```
void main()
{
    // テクスチャを参照して参照座標の画素値を取り出す
    vec4 pix = texture2D(u_texture, v_texref);
    // 画素値を反転した値を出力画素値として返す
    gl_FragColor.rgb = vec3(1) - pix.rgb; // 変更箇所(☆)
}
```

本連載では、ラズベリー・パイ5を例に、GPUプログラミングの基礎知識とコーディング手法を紹介します。

GPGPUで何か演算処理を行う場合、実現したい演算処理そのものはデバイス・コード側に記述します。今回は、前回紹介したプログラムを改造して、いくつかの基礎的な画像処理プログラムを作成してみます。

サンプル・プログラムのソースコードは、次に示す本誌ダウンロード・ページから入手できます。

<https://www.cqpub.co.jp/interface/download/contents2024.htm>

### GPGPUプログラミング…基礎編

#### ● その①…画像白黒反転プログラム

前回使用したshader-copy.hをリスト1のように

表1 基礎編で使うサンプル・コードの一覧

ディレクトリ	ファイル名	内容
texture	texture-cpu.c	ホスト・コードだけで作ったGLSL風のプログラム
	texture.c	テクスチャ描画処理を行うためのホスト・コード
	shader-copy.h	入力画像を何も加工せずにそのまま出力するシェーダ・プログラム
	shader-invert.h	入力画像を白黒反転して出力するシェーダ・プログラム
	shader-gamma.h	入力画像をガンマ変換して出力するシェーダ・プログラム
	mk.sh	ビルド用シェル・スクリプト

一部を修正して、画像反転処理用のシェーダ・プログラムを作成します。表1に示すshader-invert.hが修正後のファイルです。ホスト・コード側のtexture.cは、修正不要です。

前回も解説した通り、texture2D()関数の返り値は0~+1.0の範囲の浮動小数値に正規化されているので、1.0からテクスチャから取り出した画素値を引くだけで、画像データの白黒反転が行えます。

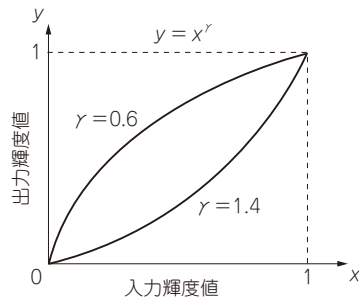
次のようにプログラムを実行すると、GPU処理により入力画像を白黒反転した画像ファイルが出力されます。



(a) 入力画像



(b) 出力画像  $\gamma = 0.6$



(c) 入出力輝度値

図1 ガンマ変換処理のイメージ

中間値の画素値を変更することで、画像全体における見た目の明るさを変化させる