

# 知っておきたい、画像データ 圧縮の仕組みと特長 ～DXTCとPVRTC～

株式会社ウェブテクノロジー  
R&Dグループ マネージャ  
小野知之  
2013.3.15

## 本講演の目的

- アプリ開発者に向けて、アプリ内で使用可能な圧縮画像フォーマットの特徴を解説します。どんな場合にどんな圧縮画像フォーマットが向いているのかが理解できるようになります。

## 前半 - 復習パート

- JPEG, PNGなどの圧縮画像の特徴についておさらいします
- ブラウザベースのアプリや、ウェブ制作者の参考にもなるように配慮しました

## 後半 - 3Dテクスチャ圧縮

- 3Dテクスチャ圧縮の特徴と注意点
- 圧縮アルゴリズム概略



# ウェブテクノロジーのご紹介

WebTechnology®  
OPTPIX imésta

OPTPIX  
imésta



OPTPIX  
SpriteStudio®



OPTPicture®



OPTPIX®  
ライブ壁紙メーカー

創業23年目になる、  
池袋のソフト開発会社です。

# 主な採用実績

## おさわり探偵 なめこ栽培キットDeluxe

© Beeworks / SUCCESS All rights reserved



## ケイオスリングス オメガ

メディア・ビジョン(株) / (株)スクウェア・エニックス  
© 2011 SQUARE ENIX CO.,LTD All Rights Reserved.  
CHARACTER DESIGN: Yusuke Naora  
Developed by Media.Vision Inc.



## まっふるマガジン キャンバスマップル(株)

Copyright(C) 2006 - 2012 MAPPLE ON Co., Ltd.  
All rights Reserved.



## 初音ミク -Project DIVA-

(株)セガ  
© SEGA / © Crypton Future Media, Inc.



## ファイナルファンタジーX-2

(株)スクウェア・エニックス  
© 2003 SQUARE ENIX CO., LTD. All Rights Reserved.  
MAIN CHARACTER DESIGN:TETSUYA NOMURA  
ALTERNATE COSTUME DESIGN:TETSU TSUKAMOTO



## 不思議のダンジョン 風来のシレン3 ～からくり屋敷の眠り姫～

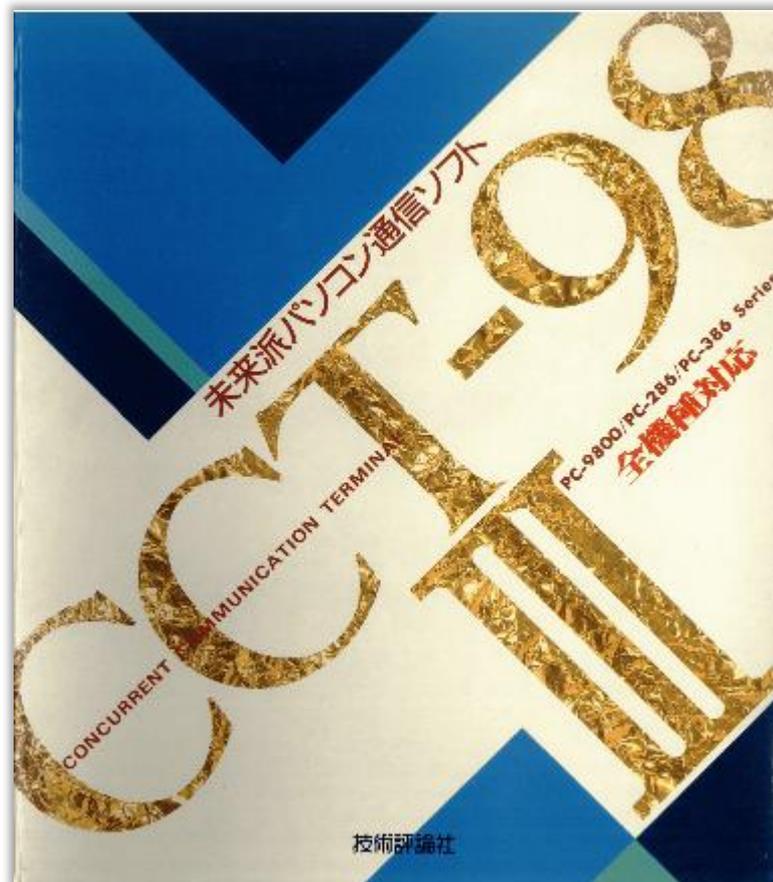
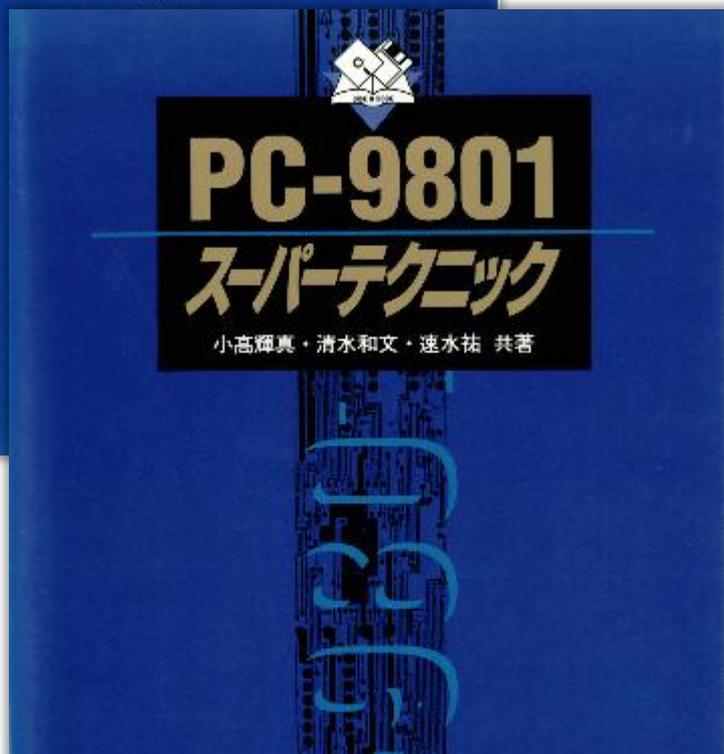
開発：(株)スパイク・チュンソフト 販売：(株)セガ  
© 2008 Spike Chunsoft Co., Ltd./すぎやまこういち



この他にも数多くのゲームおよびコンテンツ制作現場で採用



昔はこんな仕事もしていました



## 本講演の目的

- アプリ開発者に向けて、アプリ内で使用可能な圧縮画像フォーマットの特徴を解説します。どんな場合にどんな圧縮画像フォーマットが向いているのかが理解できるようになります。

## 前半 - 復習パート

- JPEG, PNGなどの圧縮画像の特徴についておさらいします
- ブラウザベースのアプリや、ウェブ制作者の参考にもなるように配慮しました

## 後半 - 3Dテクスチャ圧縮

- 3Dテクスチャ圧縮の特徴と注意点
- 圧縮アルゴリズム概略



# 復習 - JPEG

## ■ キーポイント

- 写真などの自然画の圧縮に向いている
- GUIパーツのような、くっきりした文字やアイコン主体の画像の圧縮には向いていない \*
- 画面キャプチャやバナーにも向いていない \*
- アルファ(透過度)を持たない

\* 圧縮効率という意味で



## JPEGはどうやって画像を圧縮しているのか

- 人間の視覚の特性にあわせて、情報を間引いている
- 人間の目は、明るさの変化には敏感だが、色の変化には鈍感
- 細かい変化を省く
- 8x8ピクセルごとに圧縮している。  
圧縮率を高くすると**ブロックノイズが出る**のはそのため
- サンプルング比で、色情報をどのくらい間引くかが変わる
  - 4:4:4 色情報の間引きなし
  - 4:2:2 色情報は1/2
  - 4:2:0 色情報は1/4
  - 4:1:1 色情報は1/4

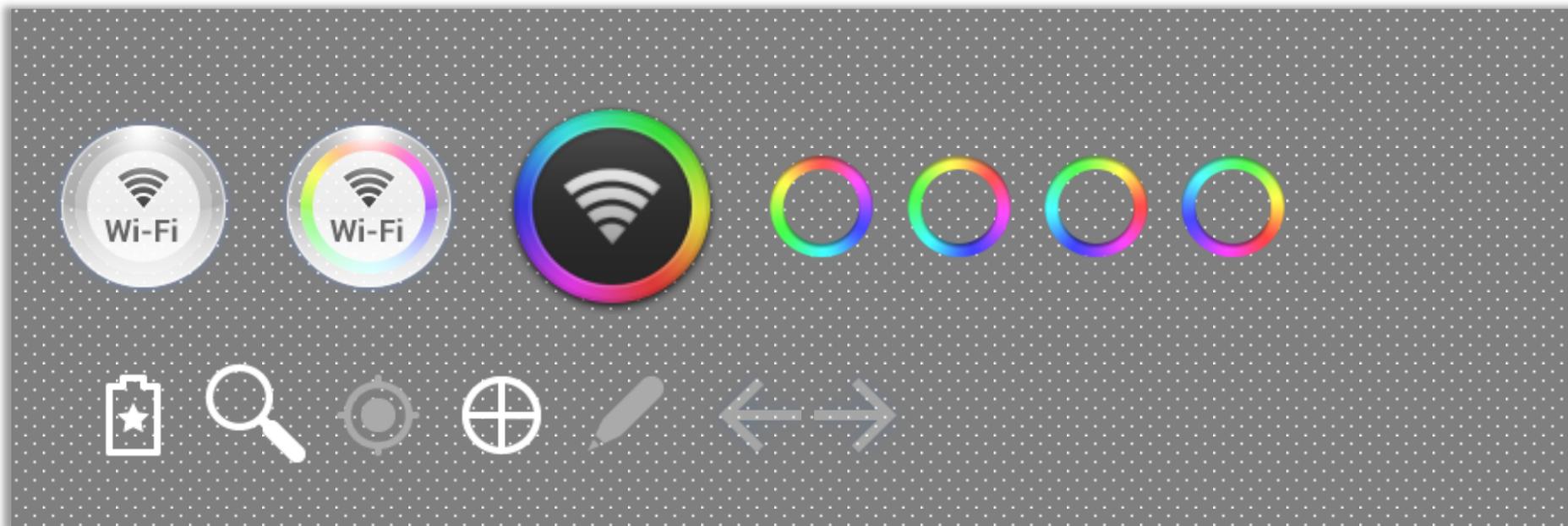


- **アプリ内で利用する意味はあまりない**  
(PNGを使う方が良い)
- ブラウザベースのアプリでも注意が必要。アニメーション可能だが、Androidのブラウザでは機種によって正しく表示できない場合がある  
(Androidの一部機種で静止GIFも正しく表示されないという報告もある)



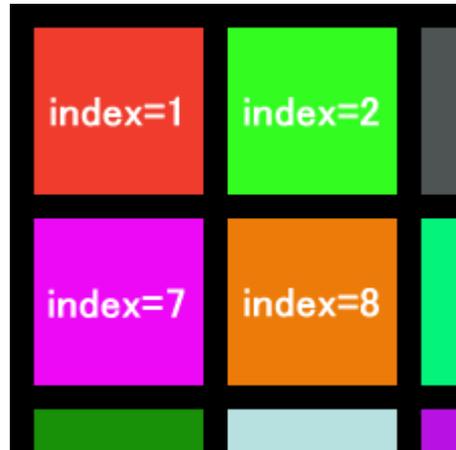
- GUIパーツのような文字・アイコン主体のくっきりした画像に向いている \*
- バナーのような、写真と文字で構成された画像は、PNG8が品質と容量のバランスが一番良い \*
- 階調アルファを持てる(美しいGUIパーツには必須)

\* 圧縮効率という意味で



# 復習 - インデックスカラーとは

## インデックスカラー



	0	1	2	3	4	5
0	0 0 0	240 60 45	51 252 33	78 84 84	54 39 237	228 237 12
8	237 123 9	3 243 123	111 12 243	12 114 24	147 21 24	24 144 8
16	21 231 225	234 231 9	234 231 255	234 24 255	234 24 114	15 24 114
24	165 21 15	229 21 15	210 42 18	255 255 255	54 240 18	54 240 150
32	54 237 18	54 51 18	18 249 255	54 219 18	54 51 255	91 25 14
40	141 7 11	7 5 234	72 47 31	53 54 53	10 62 130	115 18 107

## ■ 256×256ピクセル画像の場合

色数

$2^8=256$ 色

データ容量

$256 \times 256 \times 1 + 256 \times 3 = 66,304$  バイト

\* 画像データ自体の情報量(ヘッダ情報等は除く)



# 復習 - インデックスカラー

---

## ■ キーポイント

- 256色に減色するとデータ容量は1/3になる(アルファ付なら1/4)
- 16色に減色すれば、1/6(アルファ付なら1/8)
- アプリに組み込む際にはPNGで圧縮するのでさらに小さくなる
- 輪郭がクッキリしているので、GUIパーツやバナーに向いている
- 256色以下の色数で画像を表現するため、減色ツールの性能で画質が大きく変わる



- **Android 4.0以上ではアプリから使用可能。  
アプリ内で閉じている画像に使う場合は  
メリットあり**
- **非可逆圧縮はAndroid 3.1以降から使える  
(アンドキュメント)**
- Googleが開発した静止画フォーマット。  
**最後発なので、JPEGとPNGを合わせて高性能化したような規格**
- ブラウザでは、Chromeなど一部が対応するのみ。**普及しているとは言いがたい**
- JPEGのような非可逆圧縮、PNGのような可逆圧縮ともサポート
- 非可逆圧縮では圧縮率可変
- 階調アルファを持てる(非可逆圧縮においても)
- JPEG,PNGより圧縮率が高い



# 復習 - まとめ

## JPEG

- 写真などの自然画の圧縮に向いている
- メニューのような、くっきりした文字やアイコン主体の画像の圧縮には向いていない \*
- アルファ(透過度)は持てない

## GIF

- アプリ内で利用する意味はあまりない  
(PNGを使う方が良い)

## PNG

- メニューのような文字・アイコン主体のくっきりした画像に向いている \*
- バナーのような、写真と文字で構成された画像は、PNG8が品質と容量のバランスが一番良い \*
- 階調アルファを持てる

## WebP

- Android 4.0以上ではアプリ内で使用可能。  
アプリ内で閉じている画像に使う場合はメリットあり

\* 圧縮効率という意味で



## 参考 - JPEG の弱点

### 極端な例ですが…

PNG

- 880x50ピクセル 32色 PNG 7,993バイト (OPTPIX iméstaで減色)

コミPo! 予約受付開始! 【発売: 2010年12月15日 (水)】

予約受付

JPEG

- 880x50ピクセル JPEG 8,779バイト

コミPo! 予約受付開始! 【発売: 2010年12月15日 (水)】

予約受付



# 参考 - JPEG の弱点

PNG

- 880x200ピクセル 128色 PNG 74,484バイト  
OPTPIX imésta で減色



# 参考 - JPEG の弱点

JPEG

- 880x200ピクセル JPEG 74,708バイト



- facebookページのカバー画像は、JPEGをアップロードするとサーバー側で高圧縮率で再圧縮されてしまう
- PNGでアップロードすれば、PNGのままなので画質劣化がない  
(2013/3現在の仕様)



## 本講演の目的

- アプリ開発者に向けて、アプリ内で使用可能な圧縮画像フォーマットの特徴を解説します。どんな場合にどんな圧縮画像フォーマットが向いているのかが理解できるようになります。

## 前半 - 復習パート

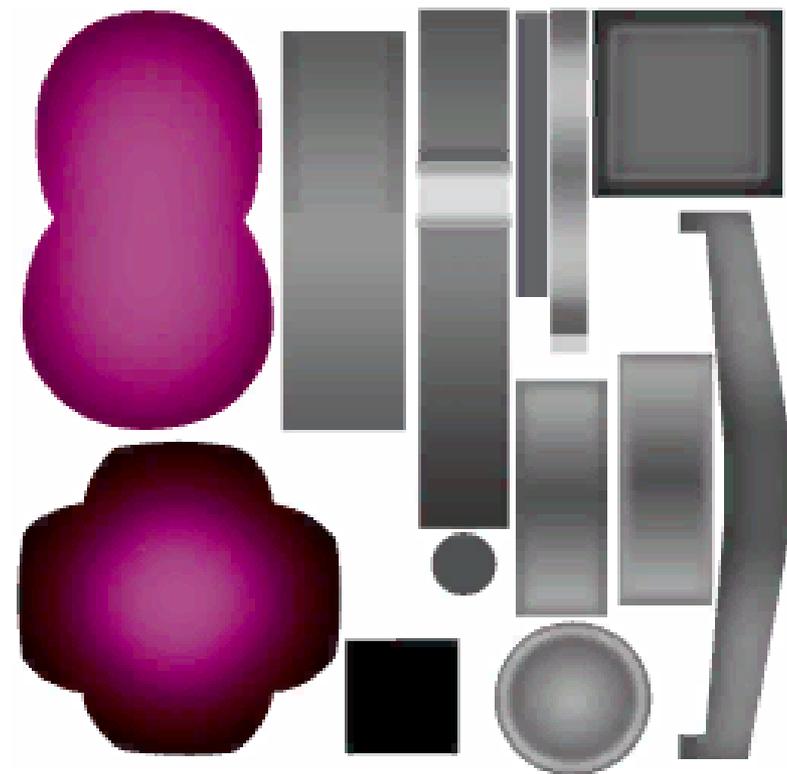
- JPEG, PNGなどの圧縮画像の特徴についておさらいします
- ブラウザベースのアプリや、ウェブ制作者の参考にもなるように配慮しました

## 後半 - 3Dテクスチャ圧縮

- 3Dテクスチャ圧縮の特徴と注意点
- 圧縮アルゴリズム概略



# 3Dテクスチャとは



# 3Dテクスチャとは



# 主な3Dテクスチャ

圧縮テクスチャの種類	変換元画像	圧縮率
インデックス カラー	RGB 24bpp	1/3(256色)*1, 1/6(16色)*2
インデックス カラー(アルファ付)	RGBA 32bpp	1/4(256色)*3, 1/8(16色) *4
DXT1(S3TC)	RGB 24bpp	1/6
DXT1(S3TC)	RGBA 32bpp	1/8
DXT2~5	RGBA 32bpp	1/4
PVRTC	RGB 24bpp	1/6(4bpp) 1/12(2bpp)
PVRTC	RGBA 32bpp	1/8(4bpp) 1/16(2bpp)
ETC	RGB 24bpp	1/6

\*1 ピクセルデータの他に、3x256バイトのパレット情報が必要

\*2 ピクセルデータの他に、3x16バイトのパレット情報が必要

\*3 ピクセルデータの他に、4x256バイトのパレット情報が必要

\*4 ピクセルデータの他に、4x16バイトのパレット情報が必要



# 主な3Dハードと主な対応テクスチャ

テクスチャの種類	1995～2000年頃	2000～2005年頃	2005～2010年頃	2010～2013年頃
インデックス カラー	PlayStation			
アルファ付 インデックス カラー		PlayStation 2		
DXTC (S3TC)		GAMECUBE Xbox PC(DirextX)	PlayStation 3 PSP Wii Xbox360 PC(DirextX)	PS4(推定) Wii U PC(DirextX) 一部のAndroid 端末
PVRTC			iPhone	iPhone/iPad PS Vita 一部のAndroid 端末
ETC				Android 端末 Nintendo 3DS

\* 各ハードウェアの発売日を表すものではありません

\* PlayStation ®、GAMECUBE ®、NINTENDO ®3DS、Wii ®、Xbox ®は各社の商標です

\* 事情により一部推定、および未記載の情報があります

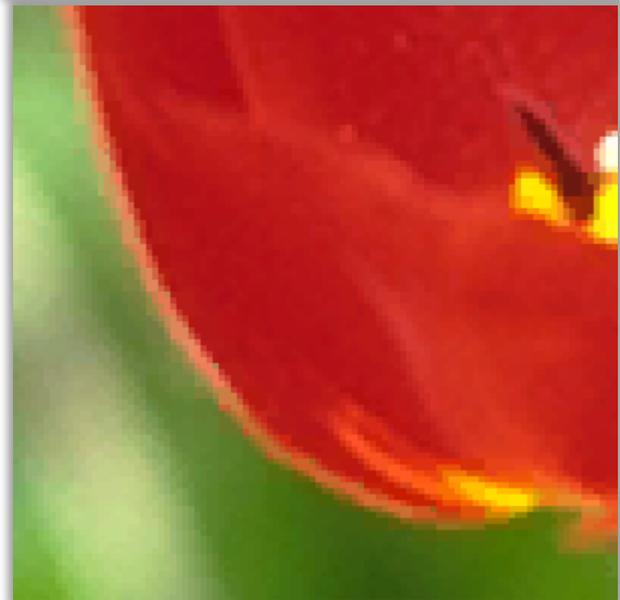


# DXTC, PVRTC の特性

DXT1画像

PVRTC (4bpp) 画像

PVRTC (2bpp) 画像

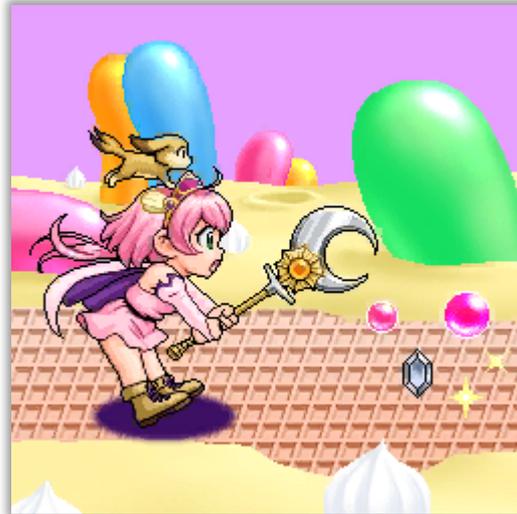


# DXTC, PVRTC の特性(弱点)

元画像(フルカラー)



DXT1画像



# DXTC, PVRTC の特性(弱点)

元画像(フルカラー)



PVRTC (4bpp) 画像



PVRTC (2bpp) 画像



## DXTC, PVRTC の特性(弱点)

元画像 (アルファ付32bit)

\* 文字の周囲が透明αで  
抜かれている画像

## DXTC, PVRTC の特性(弱点)

## 標準ツールで圧縮したPVRTC

\* 文字の周囲が透明αで  
抜かれている画像



## DirectX Texture Compression

S3TC (S3 Texture Compression) の拡張

フォーマット: DXT1, DXT2, DXT3, DXT4, DXT5

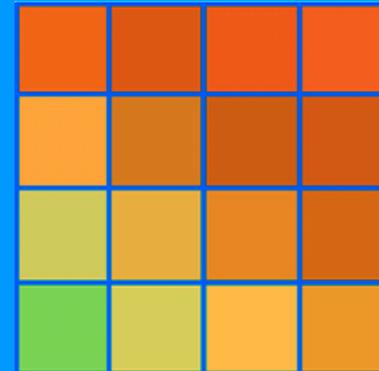
圧縮単位: 4 × 4ピクセル

PC(DirectX)やゲーム機などに広く普及



# DXTCの圧縮の仕組み (1)

4 × 4 ピクセルを圧縮の最小単位 (1 ブロック) とする

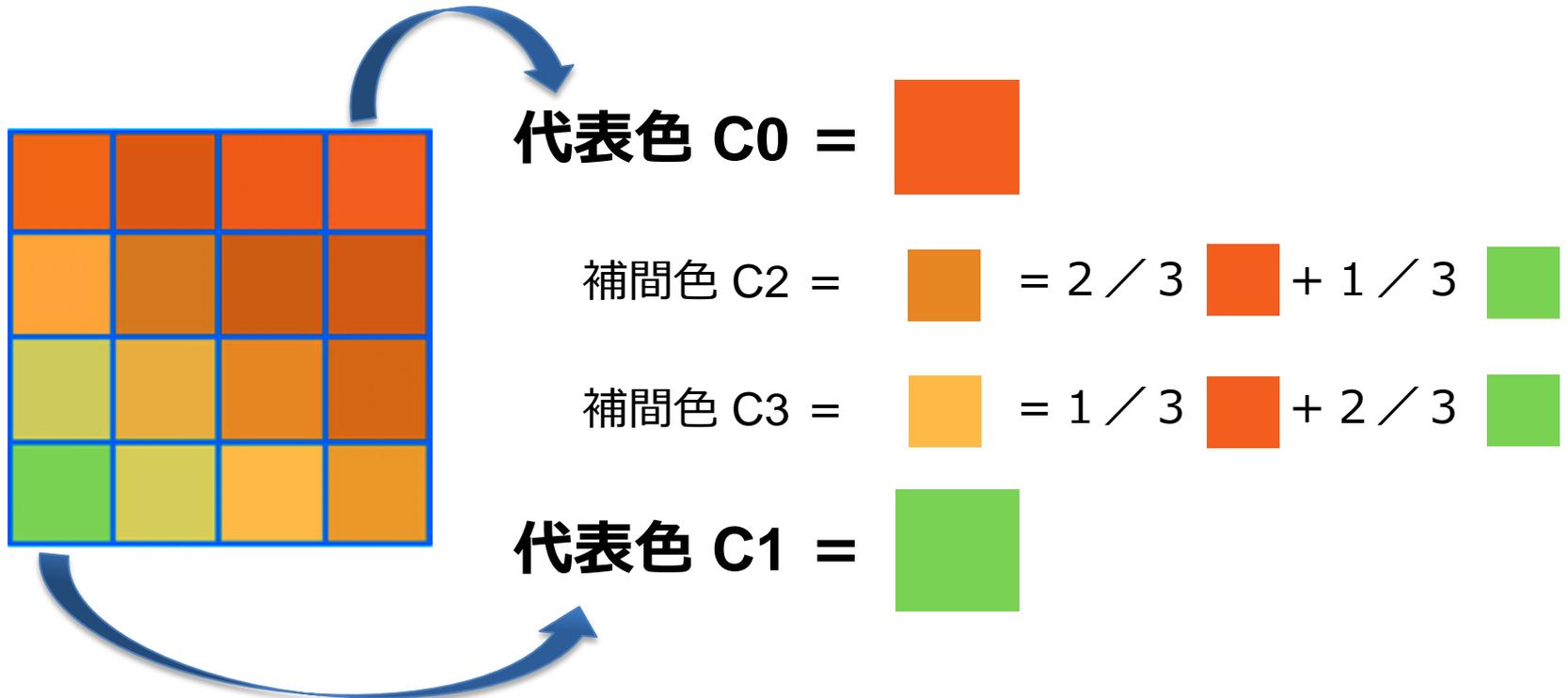


1 ブロック  
(4 × 4 ピクセル)



## DXTCの圧縮の仕組み (2)

代表色を2色選び、それらの補間色を求める



\* ブロックに無い色でも代表色に使用できます



# DXTCの圧縮の仕組み (3)

元のピクセルの色を置き換える

代表色 C0 =

0

補間色 C2 =

2

補間色 C3 =

3

代表色 C1 =

1



2	0	0	0
3	2	0	0
1	3	2	0
1	1	3	2

色にインデックス番号(0~3)を割り当て  
16個の配列データとして保存する



## インデックスと二つの色の情報のみになる

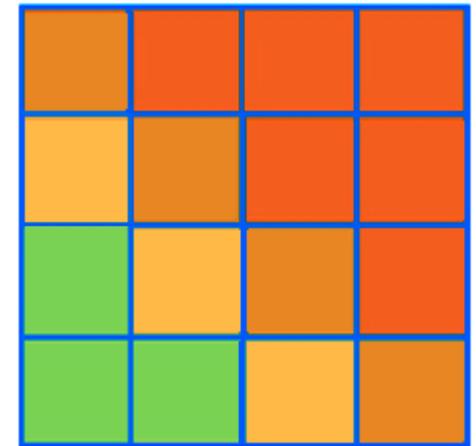
2	0	0	0
3	2	0	0
1	3	2	0
1	1	3	2

インデックス2ビット  
× 16ピクセル  
= **32 ビット**

代表色 C0 =   
代表色 C1 = 

16ビット (RGB565)  
× 2色  
= **32 ビット**

復元

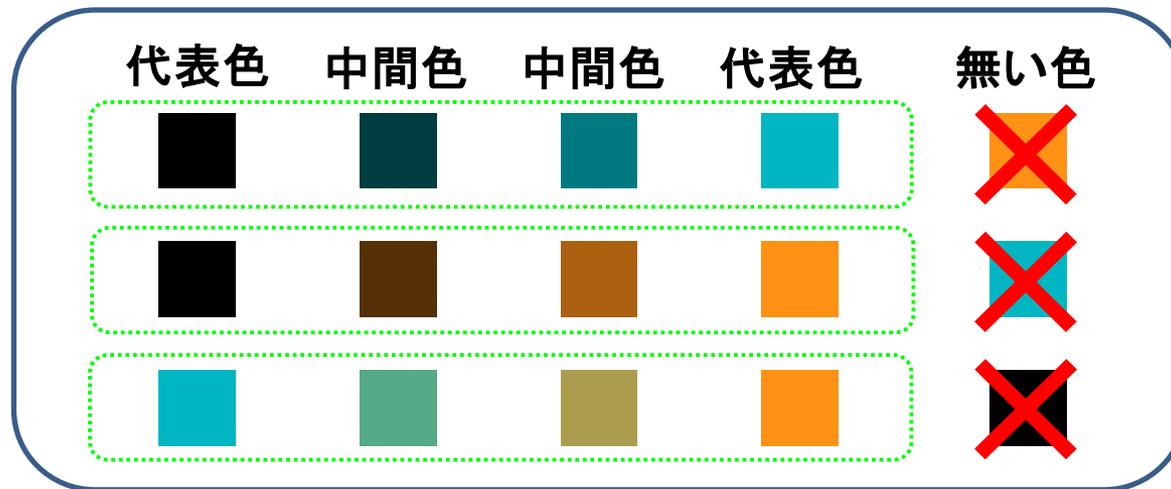
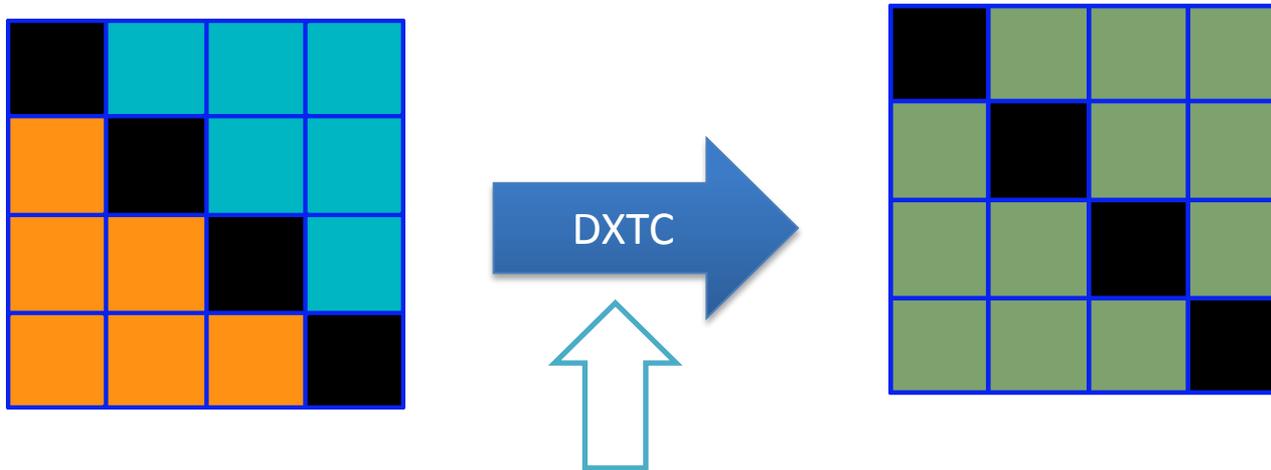


1ブロックあたり 64 ビット

\* DXT2~5のアルファチャンネルはこれと異なります。



## DXTCで表現できないパターン



## 補足 - DXT1の補間モード

## 代表色の順序で補間方法が変わる

代表色  $C_0 > C_1$  $C_0 = \#F35D1E$  $C_1 = \#7AD254$ 

**補間2色**
代表色  $C_0 =$ 補間色  $C_2 =$  $= 2 / 3$  $+ 1 / 3$ 補間色  $C_3 =$  $= 1 / 3$  $+ 2 / 3$ 代表色  $C_1 =$ 代表色  $C_0 \leq C_1$  $C_0 = \#7AD254$  $C_1 = \#F35D1E$ 

**補間1色  
+透過色**
代表色  $C_0 =$ 補間色  $C_2 =$  $= 1 / 2$  $+ 1 / 2$ 代表色  $C_1 =$ 透過色  $C_3 =$ 

# DXTCのフォーマット

名前	RGBチャンネル	アルファチャンネル	圧縮率*	乗算済みアルファ
DXT1 (S3TC)	代表 2 値 + 補間 2 値 or 1 値	未使用 or 1ビット(抜き)	1/8**	
DXT2	代表 2 値 + 補間 2 値	4 ビット(16階調) 固定アルファ値	1/4	○
DXT3	代表 2 値 + 補間 2 値	4 ビット(16階調) 固定アルファ値	1/4	
DXT4	代表 2 値 + 補間 2 値	補間(8階調) アルファ値	1/4	○
DXT5	代表 2 値 + 補間 2 値	補間(8階調) アルファ値	1/4	

\*32ビット非圧縮ARGB画像からの圧縮率

\*\*24ビット非圧縮RGB画像からの圧縮率は1/6



## PowerVR Texture Compression

フォーマット: 4bpp, 2bpp

圧縮単位:  $4 \times 4$ ピクセル

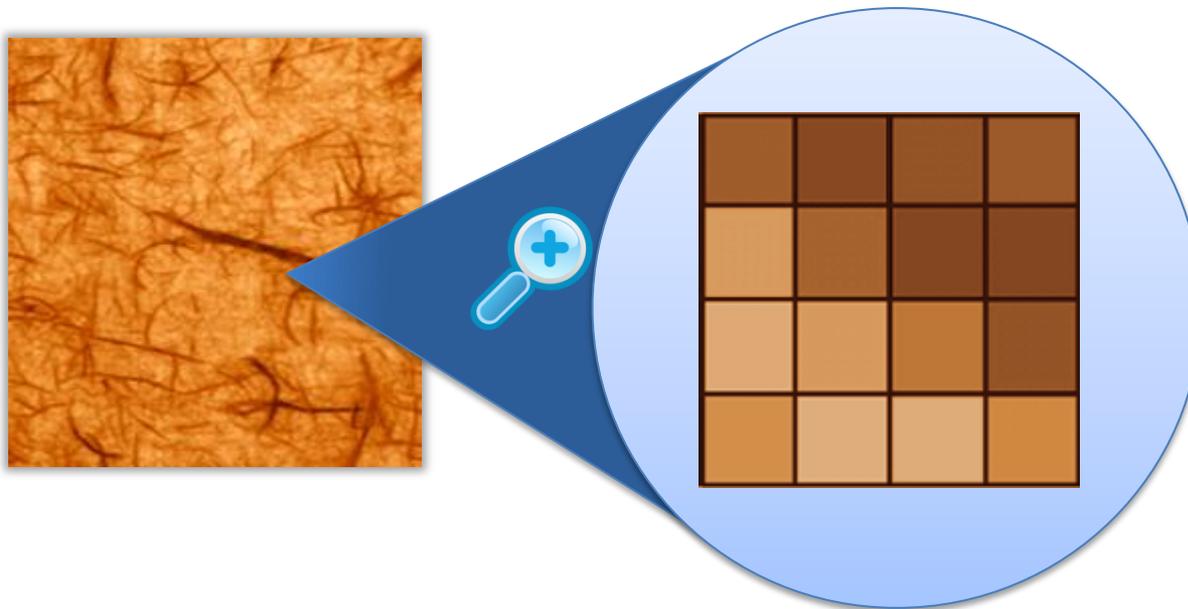
画像の縦横サイズは2の累乗の正方形(256x256等)のみ

スマートフォンでの利用が拡大



# PVRTCの圧縮の仕組み (1)

4 × 4 ピクセルを圧縮の最小単位(1ブロック)とする

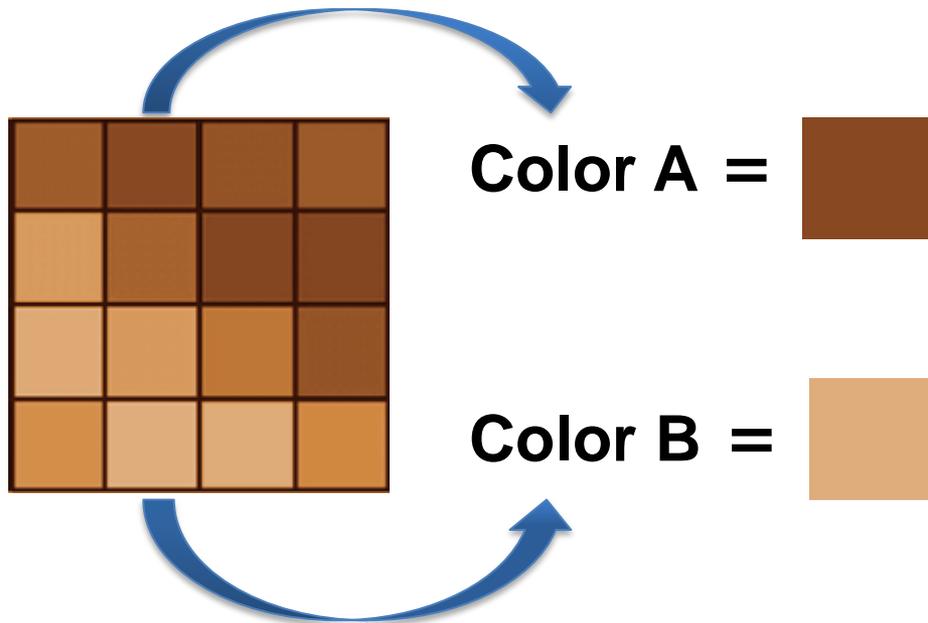


1 ブロック  
(4 × 4 ピクセル)



## PVRTCの圧縮の仕組み (2)

### ブロックごとに2色を決める

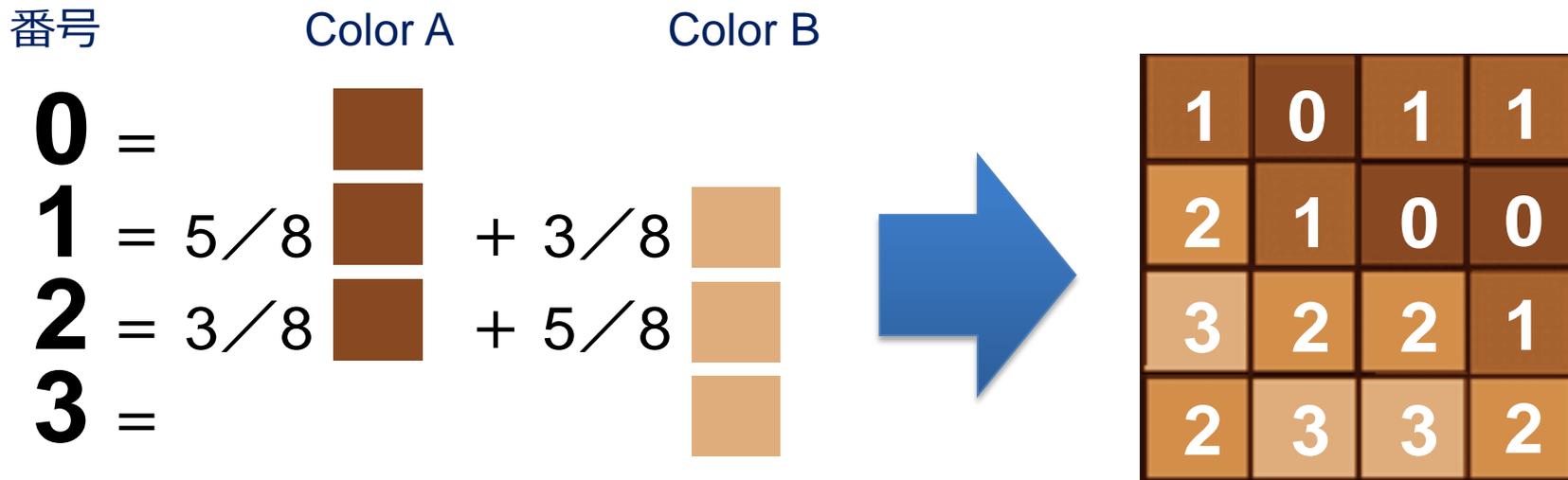


- \* ブロックに無い色でも指定できます
- \* 実際には複雑な方法で2色を選ぶ必要があります (詳細は後述)



## PVRTCの圧縮の仕組み (3)

ピクセルの色を、ブレンド方法の番号に置き換える



この数値を Modulation Data と呼び、  
配列データとして保存する



# PVRTC (4bpp) のデータ構造

## 2色とModulation Dataの情報になる

Color A =  = 16 ビット

Color B =  = 15 ビット

Mode Bit = 1 ビット

1	0	1	1
2	1	0	0
3	2	2	1
2	3	3	2

Modulation Data  
2ビット × 16ピクセル  
= 32 ビット

復元方法に  
特徴あり

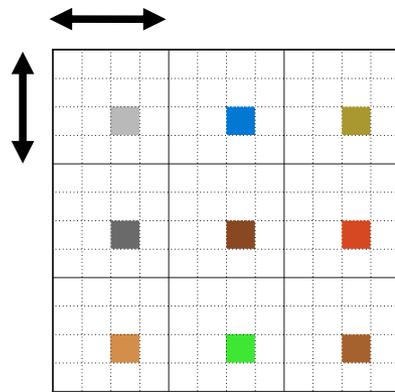
1ブロックあたり 64 ビット



# PVRTCの復元 (1)

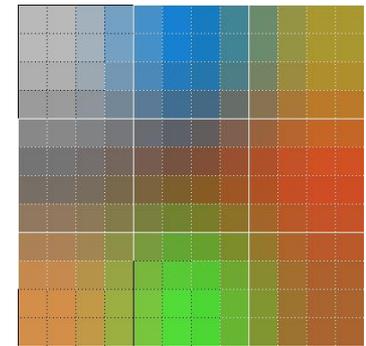
Color A, B をそれぞれ、周囲のブロックと補間する

1ブロック(4×4)



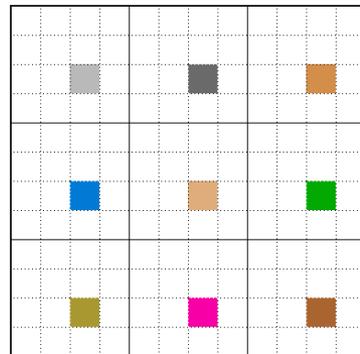
Color A

補間

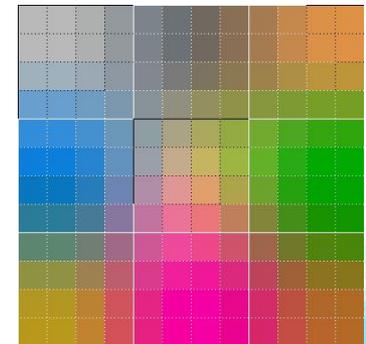


4×4ピクセルを補間色で埋める

Color B

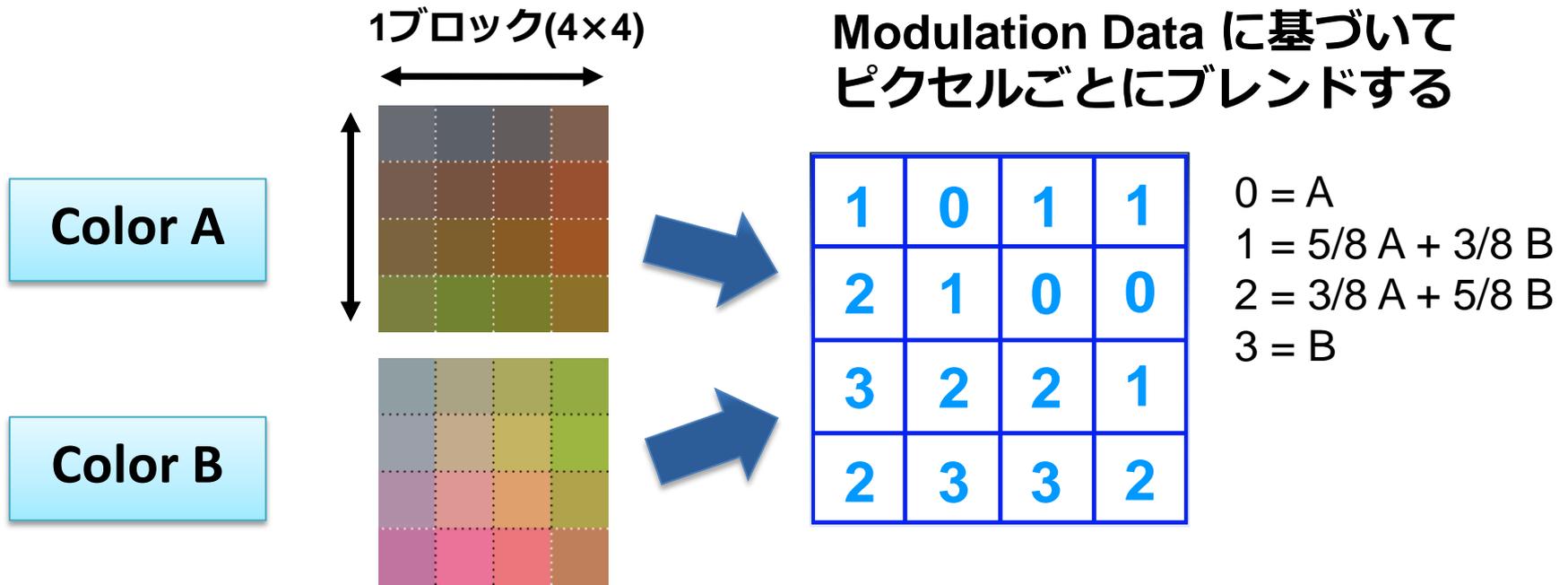


補間



# PVRTCの復元 (2)

## 補間結果のピクセルごとにブレンドする



補間結果を想定した Color A, B を、  
圧縮時に適切に選んでおく必要がある



## 補足 - PVRTC(4bpp)のモード

## Mode Bit の値でブレンド方法が変わる

Mode Bit = 1

➔ 4段階

番号	Color A		Color B
0 =			
1 =		+ 3/8	
2 =		+ 5/8	
3 =			

Mode Bit = 0

➔ 3段階  
+透過

番号	Color A		Color B	
0 =				
1 =		+ 4/8		
2 =		+ 4/8		& Alpha = 0 
3 =				



## 補足 - PVRTC(4bpp)の色形式

## アルファの有無を選択可能

**Color A** = 16 ビット

アルファなし: RGB555

アルファあり: ARGB3444

先頭ビットで切り替え

1	Red 5 bit	Green 5 bit	Blue 5 bit
---	--------------	----------------	---------------

0	Alpha 3 bit	Red 4 bit	Green 4 bit	Blue 4 bit
---	----------------	--------------	----------------	---------------

**Color B** = 15 ビット

アルファなし: RGB554

アルファあり: ARGB3443

1	Red 5 bit	Green 5 bit	Blue 4 bit
---	--------------	----------------	---------------

0	Alpha 3 bit	Red 4 bit	Green 4 bit	Blue 3 bit
---	----------------	--------------	----------------	---------------



# まとめ - DXTC, PVRTC の特性

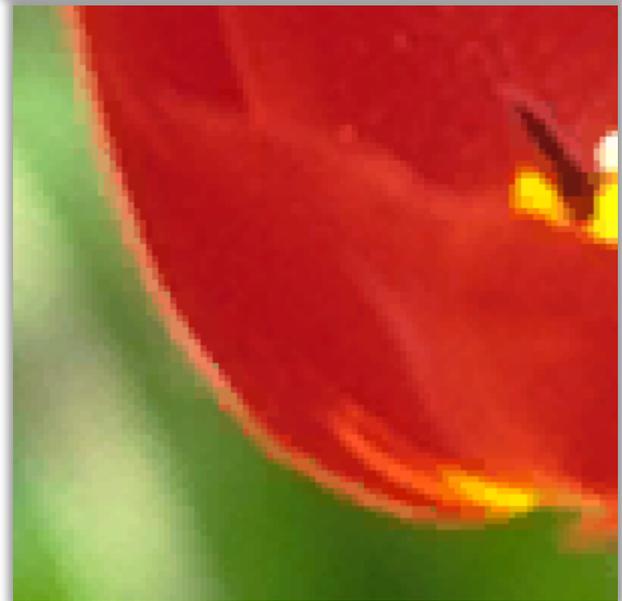
DXTC1画像



PVRTC (4bpp) 画像



PVRTC (2bpp) 画像



# まとめ - DXTC の弱点

元画像(フルカラー)



DXT1画像



# まとめ - PVRTC の弱点

元画像(フルカラー)



PVRTC (4bpp) 画像



PVRTC (2bpp) 画像



# まとめ - DXTC, PVRTC の特性

写真のようなテクスチャは劣化が少ない

線画や文字、GUIパーツなど、  
クッキリしたテクスチャの圧縮には不向き

解決  
方法

圧縮の特性を理解して元絵を作る

(手戻りを無くすために開発初期に圧縮画質を評価する)

インデックスカラーを使う

(ただし、描画速度的には不利)



# 最後に少しPR



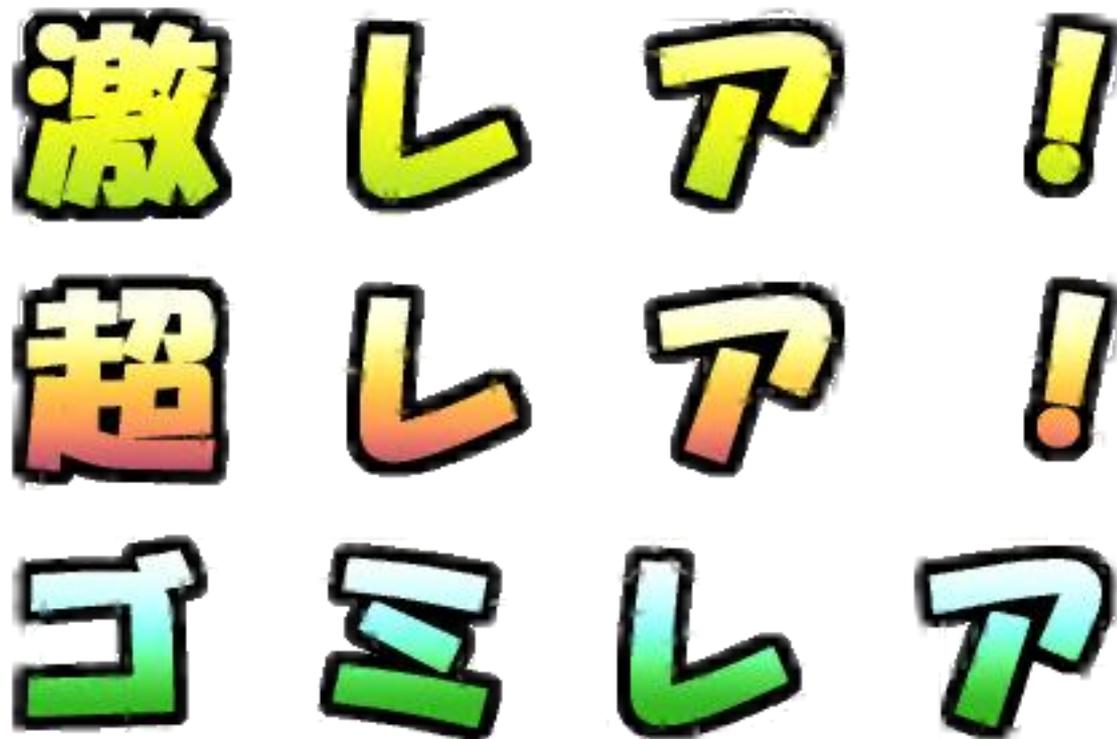
元画像 (アルファ付32bit)

\* 文字の周囲が透明αで  
抜かれている画像



## 標準ツールで圧縮したPVRTC

\* 文字の周囲が透明αで  
抜かれている画像



## OPTPIX iméstaで圧縮したPVRTC

\* 文字の周囲が透明αで  
抜かれている画像



\* 文字の周囲をアルファで抜いたような画像に高い効果があります



# OPTPIX imésta 7

for Mobile & Social

PNG

JPEG

GIF

WebP

Index  
Color  
(CLUT)

DXTC  
(S3TC)

PVRTC

ETC



クリエイターの笑顔のために、  
皆様の困ったことを解決するのが  
ウェブテクノロジーの仕事です。



御清聴ありがとうございました

<https://www.facebook.com/OPTPIX>

<http://www.webtech.co.jp/>

