

Alpha Compositing - Part II

開始日	2007年03月17日
最終更新日	2009年06月08日

はじめに

[Alpha Compositing - Part I](#)で説明した方法を応用し、Web2.0的な画像を作成する方法について説明・使い方を記しています。
全4ページのうち、ここは1ページ目です。

原著

「Alpha Compositing - Part II」
<http://rmagick.rubyforge.org/web2/web2.html>

注意

- もともと個人利用を目的として日本語化したために、けっこう意識している部分があります。「意味分からないよ」とか「おかしいんじゃない?」とかいうのがあれば、オリジナルを参照するか、コメントで質問してください(がんばって調べます)。

用語

訳文に出てくる各語に対応する原文と意味を以下に記します。

訳語	原文	意味
元画像	source image	合成しようとしている画像。
合成先画像	destination image	元画像の下に重ねる画像。

更新履歴

- 2008/3/17 作成開始(Part1から1年以上経ってしまった・・・。しかもWeb2.0ってブーム過ぎてるし)
- 2008/3/19 とりあえず作成完了。あとはチマチマ微修正。

訳文

Tommy Maloneyの[Web 2.0 Design Kit](#)は、Photoshopを使ってWeb2.0チックなエフェクトを作成するチュートリアル記事が書かれたサイトだ。私はこのチュートリアルを試してみて、すぐに気付いたよ。自分の書いたアルファコンポジットのPart2を書かなきゃってね(まだ読んでないなら、[ここ](#)を読んでくれ)。

このチュートリアルでは、Part1で紹介したエフェクトをもう1度使って、PhotoshopではなくRMagickによるWeb 2.0 Design Kitの説明をする。でも、できたら始める前に[本家のページ](#)を一通りこなし、

チュートリアル内で画像が作成される様子を見ておいてほしい。RMagickバージョンの完成画像は、下のイメージのようにちょっと違っているからね。



以降では、このイメージをRMagickでどうやって作成していくかを紹介する。そうだ、Part1で説明したコンポジット(composite)に加え、テキストを描画するためにDrawクラスからannotateを、背景のグラデーションにはGradientFillクラスを、そして影の作成にblur_imageを使う。そしてappend、rotate、negateメソッドもだ。

グラス効果(The glass effect)

最初のエフェクトは反射、いや「グラス」エフェクトだ。始める前に、作業場を片付けておくんだ。そしてImageListオブジェクトに貯め込んでいく画像や、グラデーションの大きさを決める2つの定数を定義してくれ。

```
require 'RMagick'  
include Magick  
  
WIDTH = 650  
HEIGHT = 40  
  
stripes = ImageList.new
```

それじゃ、背景にブルーグレイのストライプを作るところから始めよう。こいつは2つのグレイスケールのグラデーションを組み合わせることで出来る。2つのグラデーションを持つ画像を上下に重ねて1つの画像にして、compositeメソッドのColorizeCompositeOpで色を重ねるだけだ。

グラデーション属性を、GradientFillクラスの2つのインスタンスによって作ろう。GradientFill.newメソッドには6個の引数が必要だ。最初の4つは直線か座標に使われ、残りの2つはグラデーションの色を決める。そうするとグラデーション属性として、最初の直線(or座標)から1つ目の色で始まり、2つ目に指定した色に残る直線(or座標)へと変化しながら出来上がる。

色：#ddddddで画像上端の線から、色：#888888で画像下端の線に向かうグラデーションを作る。そしてそれをImage.newの第3引数、つまりfill属性に渡すことで、画像が出来上がる、というわけだ。出来上がった画像はstripesに突っ込んでおいてくれ。

```
top_grad = GradientFill.new(0, 0, WIDTH, 0,  
    "#dddddd", "#888888")  
stripes << Image.new(WIDTH, HEIGHT, top_grad)
```

次にもう少し暗い色でグラデーション属性を作り、画像を作る。stripesに追加するのも同じだよ。

```
bottom_grad = GradientFill.new(0, 0, WIDTH, 0,  
    "#757575", "#555555")  
stripes << Image.new(WIDTH, HEIGHT, bottom_grad)
```

さあ、画像を見てみよう。まずこれが上端の画像だ。



こちらは下端の画像。



この2つをくっつけて、1つの画像を作る。appendメソッドはtrueかfalseかの1引数のみ取る。trueだと画像は上下に重ねられ、falseだと左右に並べられるよ。

```
combined_grad = stripes.append(true)
```

今回はtrueだから上下に重ねられる。そして、結合後の画像がこれだ。



色の合成(The colorize composite operation)

いい感じになってきた。

さて次は色味を与えよう。グレースケールのグラデーションに色味を与える場合はcompositeメソッドを使うよ。Part1で、アルファコンポジットには元画像と合成先画像の2つを使ったのを覚えているかい？今回は、合成先画像にたった今作ったグラデーション画像を使うんだ。だから、元画像をこれから作る。これで画像に色味が加えられるわけだ。

合成するグラデーション画像と同じサイズの元画像を作ってほしい。ソリッドな青色の背景だね。

```
color = Image.new(combined_grad.columns, combined_grad.rows) do
  self.background_color = "#87a5ff"
end
```

この画像を表示すると、こうなっている。



ところで、Part1で説明したsrc-overによる合成は、元画像のピクセルを合成先の画像で置き換えてしまう。これだとちょっとマズいのは分かるよね。せっかく作ったグラデーションが青一色になってしまうから。今回はそうでなく、元画像と合成先画像とをブレンドしたい。そうするには、色のHue(色相)、Saturation(彩度)、Brightness(明度) *1 という用語について知っておくといいよ。

僕らは皆、色といえば赤/緑/青の組み合わせで考えるよね。これはRGB系というんだ。同様に、色相/彩度/明度の組み合わせ：HSB系も存在する。このHSB系では、色相(H)は色相環における場所を示し、明度は色がどのくらいの明るさを持っているかを示す。分かりやすい例で言えば、黒は明度0だし白は明度100(%)だね。そして彩度は鮮やかを示すんだ。

RMagickはHSL系で表された2画像間のピクセルをフルレンジで合成することが出来るんだ。用いる方法はcolorize。colorizeによる合成は、「元画像の色相と彩度」と「合成先画像の明度」を合成するよ。

さて、思い出してくれ。RMagickにおいて、合成先画像とはcompositeメソッドを送信する元のことだ。だから元画像が第1引数になる。2つ目の引数は、元画像の合成先画像に対する相対的な位置を示すわ

けだけど、今回は2つとも同じサイズなわけだ。さて何を使ったら良いだろう？ま、ここでは後で分かりやすくするためにCenterGravityを選んでおいたよ。
そして3つ目の引数で、どうやって合成するかを示してくれ、ColorizeCompositeOpだね。

```
background = combined_grad.composite(color,  
    CenterGravity, ColorizeCompositeOp)
```

色味の着いたグラデーションはこんな感じになるよ。



次は"RMAGICK"という単語を反射エフェクト付きで埋め込んでいく。

Page 2: 反射するテキストの追加

参考サイト(訳者による追加)

- RMagickオンラインドキュメント：<http://www.simplesystems.org/RMagick/doc/index.html>
 - compositeメソッド：<http://www.simplesystems.org/RMagick/doc/image1.html#composite>
 - CompositeOperator定数の一覧（src-over以外にも多数あり）：<http://www.simplesystems.org/RMagick/doc/constants.html#CompositeOperator>
 - compositeメソッド（Drawクラス）：<http://www.simplesystems.org/RMagick/doc/draw.html#composite>
 - RVGのチュートリアル：<http://www.simplesystems.org/RMagick/doc/rvgtut.html>

(11683)

コメント

コメント大歓迎です。ただ、スパムなど内容によっては独断で削除する場合があります。あらかじめご了承ください。

名前:	<input type="text"/>
コメント:	<input type="text"/>

投稿