イメージマジックの例 - ビデオ処理

インデックス

ImageMagickはデジタルビデオの取り扱いに特に適していません 処理しますが、特にLinuxでは、この目的で一般的に使用されます 環境。ここでは、実際の取り扱いに固有のテクニックと例を探ります ライフ(およびレイトレーシング)ビデオシーケンス。

ビデオからGIFへ、最適化の概要

IMを使用してムービーGIFを作成するソフトウェア開発者、ブノワ・ルーロー、ディスカッション中 私と一緒に、飛行機の上空を飛んでいるAVIビデオをくれました、 IMビデオ変換技術を相互に探求するのに役立ちます。ただし、AVI自体は非常に小さいですが、非圧縮ビデオ はサイズが巨大な`6201996`バイトであり、フレーム上の`107`色が含まれます`32448`。

ただし、IMはこのビデオをGIFアニメーションに変換するのに問題はありません。ただし、サポートされていない「AVIチャンク」を取得する可能性があることに注意してください エラーは、"" コントロール設定を使用して無視できます。`-quiet`

```
magick -quiet -delay 1 plane.avi plane.gif
```

これは、ImageMagickのデフォルトのカラー量子化を使用しました ビデオの非常に合理的な変換を生成するためのディザリング方法。ビデオは非常に少ない色を使用するため、色の問題はほとんどありません。 端を発する。特にGIFには256色があるため、これが常に当てはまるとは限りません フレーム制限ごと。ただし、アニメーションファイルのサイズは`1077859`バイトですが、減色とGIFのため、サイズはわずか1/5ですが ピクセルデータ圧縮、それはまだかなり大きいです。また、結果のアニメーションをさらに調べると、画像内のフレームのうち、`106`フレームには独自の個別のローカルがあることがわかります。`107` カラーテーブルが追加されました。つまり、GIFアニメーションのすべてのフレーム 独自のカラーインデックステーブルが必要です。つまり、各フレームはそれが少ないですが 256色(GIF形式の制限により)、アニメーション全体が `7525`色の合計。残念ながら、GIF形式はカラーテーブルを圧縮しないため、それらすべて 追加のカラーテーブルは、最大256色*色あたり3バイトまで使用できます * 106フレーム;または 81,408 バイトのファイル領域。1Gバイトではそれほど多くない ビデオですが、特に最適化すると、かなりの量のスペースがあります さらにビデオ。これに加えて、アニメーションはGIFフレームの最適化がうまく行われません。じゃない背景が動いているという理由だけで(カメラが上向きにパンするため)、しかし また、IMがエラー訂正を使用したため ディザ(ヒルベルト曲線ディザ)、これは擬似ランダムパターンを生成します フレームごとに異なる色。後の例では、これを作成します「ディザノイズ」がはるかに目立ちます。

共通グローバルカラーテーブル

ここでは、単一のグローバルカラーを生成します ビデオのすべてのフレームの表。

```
magick -quiet -delay 1 plane.avi  +remap   plane_cgc.gif
```

これにより、当然`0`、ローカルカラーテーブルとファイルサイズが`1060471`バイトになります。

ご覧のとおり、結果のアニメーションには追加のローカルカラーテーブルはありません。代わりに、IM は、アニメーション内のすべてのフレームに基づいて、「最適な」色の単一のグローバル カラー テーブル`256`を生成しました。残念ながら、これによりピクセルデータが圧縮されず、より強いディザが必要だったので、以前はそうでした。結果はわずかに 見栄えの悪いアニメーション、それは前のものとほぼ同じサイズです。限られた色のこの特定のビデオのために、私は数を減らすことさえできます さらに使用される色は、あまり問題なく64色だけと言いますが、さらに小さいアニメーションファイルサイズを生成します。しかし、これは非常に依存しています 使用されているビデオシーケンスでは、あまり見栄えが良くない場合があります。自分の動画は、特に次の場合に、より良い結果またはより悪い結果をもたらす可能性があります より多くの色とおそらく複数のシーンを使用するビデオを処理します。

ユニバーサルグローバルカラーテーブル

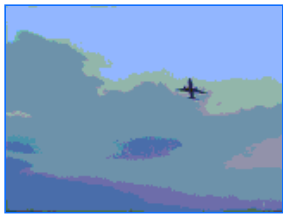
「小さな」GIFアニメーションを生成するより良い方法は、一般的な「最高の」グローバルカラーテーブルを生成するのではなく、普遍的な色の範囲 アニメーション。存在する色に関係なくうまく機能するはずのものを使用してください 元のビデオで。これを行うもう一つの理由は、あなたがなしであなたをより長くすることができるということです 色の選択に深刻な悪影響を及ぼしたり、ローカルカラーを並べ替えたりします 各フレームのテーブル。各フレームは同じカラーマップにディザリングされ、アニメーション内の他のフレームとは完全に無関係です。

ここでは、'332'カラーマップを使用します。通常、透明度がない場合に非常に優れた標準カラーマップと見なされます が必要です。私はこのカラーマップ(または219色の「Webセーフ」カラーマップ)がさまざまなものでよく使用されているのをよく見てきました。ビデオフォーマット。

```
magick -quiet -delay 1 plane.avi -remap colormap_332.png plane_ugc.gif
```

このアニメーションにはローカルカラーテーブルがあり`0`、その結果、アニメーションのサイズが小さくなるかバイト`704091`になります。ただし、問題は、明白で迷惑な「ノイズ」が頻繁に表示されることです。 一定の色の領域で。このノイズは、以前のすべてにも存在していました ビデオアニメーション。より普遍的なものを使用しているため、今だけ見えています。したがって、カラーマッピングをより広く広げます。ノイズは、実際には、次の場合に設定された縮小カラーのディザリングによって引き起こされます。 イメージを再生成します。ただし、これにより、 フレームごとに変化する色により、 画像内のバックグラウンドノイズ。これが発生する理由の詳細については、E-Dithersの問題を参照してください。

カラーディザリングをオフにして、「ディザノイズ」を取り除くことができます...



```
magick -quiet -delay 1 plane.avi \
        +dither -remap colormap_332.png plane_ugc_nd.gif
```

ローカルカラーテーブルがあり`0`、サイズはバイトです`110866`。

結果として得られるアニメーションは、元の1/60のサイズが非常に小さいです アニメーション、一般的に単色生成の広がりのため 非常に優れたピクセル圧縮。しかし、それはディザノイズを修正しながら、そして 非常に小さなファイルサイズにすると、代わりにカラーバンディングが得られます。 一般的に非常に悪いトレードオフと見なされています。

注文されたディザリングビデオ

本当の解決策は、別のカラーディザリング技術を使用することです。あるフレームから次のフレームに異なるパターンを生成しません。

たとえば、ここでは順序付きディザを使用しました ポスタリゼーションされたカラーレベルを使用して、同じユニバーサル「332」カラーマップをディザリングします。

```
magick -quiet -delay 1 plane.avi \
        -ordered-dither o8x8,8,8,4 +remap plane_od.gif
```

ローカルカラーテーブルがあり`0`、サイズはバイトです`303037`。

上記では「」も使用しました 演算子を使用して、すべての画像がまったく同じグローバルカラーマップを使用するようにします (注文されたディザはすでに最大256色に減少しています)。として 色数はすでに最適であり、""演算子はディザリングや減色を行いません。結果のディザパターンはランダムではなく、あるフレームから次のフレームへ。したがって、「ディザノイズ」は アニメーションにより、fromからフレームへの固定カラーパターンが得られます。パターンも非常に反復的であり、はるかに優れた圧縮を可能にします。そして最後に、カラーマップが修正されているので、それはかなりうまくいくはずです 使用されているビデオに関係なく。+remap+remap

高品質の注文ディザリングビデオ

ただし、この特定のビデオでは、ほとんどがさまざまな色の範囲しか使用していません。 青の色合いなので、実際にはによって提供される色をあまり使用しません 一般的なユニフォームカラーマップ。実際、最後のビデオアニメーションでは色だけが`31`使用されていました!これは非常に低く、それ自体もかなり目立ちます。しかし、それはまた、この特定のアニメーションは、多数の「色」を使用することで恩恵を受けることができます 順序付けられたディザ操作のレベルは、全体的な品質を向上させます。ただし、最初に、アニメーションがいくつのカラーレベルを取得できるかを決定する必要があります 両方のGIFファイルによって課せられた256色の制限に達する前に処理します フォーマットとグローバルカラーマップの再マッピング。ただし、トリッキーな部分は、保存する前にこれらを決定する必要があることです。 限られたGIF形式へのアニメーション。そして、ここに私が使用するコマンドがあります...

```
magick -quiet plane.avi -ordered-dither o8x8,23 -append -format %k info:
```

```
235
```

基本的に、使用するカラーレベルの数を増減しました。 必要な256色の制限内にちょうど収まった数字を持っていました。

次に、検出された「カラーレベル」の選択を平面アニメーションに適用できます。

```
magick -quiet -delay 1 plane.avi \
        -ordered-dither o8x8,23 +remap plane_od2.gif
```

ローカルカラーテーブルがあり`0`、サイズはバイト数と`235`色です`660614`。

ご覧のとおり、非常に高品質の注文ディザビデオが生成され、これは、「最高のカラーマップ」グローバルカラーマップバージョンと同等です。 以前に生成されましたが、サイズは1/3小さく、「ディザノイズ」は 今でははるかに見づらいです。もちろん、品質がはるかに高いため、より大きなファイルが必要です 低品質バージョンほど圧縮されないため、サイズ。一方、あなたは実際に品質をうまく制御できるようになりました vs 使用される「カラーレベル」の数の形でのファイルサイズのトレードオフ。このテクニックは、アニメーションにとって*特別なケース*であることを覚えておいてください。 あまり多くの色を使用しません。そして、さらに追加してビデオを長くする フレームはまた、より多くの色を追加するため、色を減らす必要があります レベルの品質管理。これは私がまだ見たことのない色最適化の最良の方法についてです 一般的なGIFアニメーション。「ディザノイズ」を除去し、ある程度の品質を提供します コントロール、および他のGIFアニメーション最適化を使用する機能を保持します フレーム最適化などのメソッド。

圧縮(透明度)の最適化

この動画はパンニングカメラを使用しているため、動画の背景は フレームごとに変化します。これは、GIFアニメーションがフレーム最適化をうまく行わないことを意味します。

ただし、単純な透明度の最適化を使用して、GIFの最終サイズをさらに縮小することはできます。 アニメーション。

```
magick plane_od2.gif  -layers OptimizeTransparency +remap plane_opt.gif
```

結果は、バイトのサイズと`236`色です`531981`。

すなわち、1つの余分な色、透明なカラーインデックスを画像に追加し、現在表示されている色を変更しないピクセルが作成されました 透明。これにより、透明な領域の大きなセグメントが生成されます。 元のアニメーション、および同様のピクセルシーケンスの繰り返し。 は、最終的な GIF 画像で改善された LZW 圧縮を生成します。悪くない、アニメーションはGIFへの直接変換の半分になりました、それでも適度に高品質です。*上記に追加したい場合は、さらに改善するためのテクニックについて話し合ってください それら、私、またはIMフォーラムに連絡してください。私は聞いてとてもうれしいです あなたの見解、テクニック、ディスカッション、または特定のビデオ/アニメーションを見る あなたが持っているかもしれない問題。*そのような議論の1つは、アニメーションGIFで量子化するための「適切なレベル」を見つけることです。

ギフロッシー圧縮LZW最適化

新しいツール、GifLossyこれは 元のGifsicleプログラムのフォークは、LZWが圧縮できるように各フレームの色を変更します はるかに多くの画像。

たとえば、ここでは元のGIFアニメーションに適用し、次のように依頼しました。 色を 1 つの 256 色のテーブルに減らします。

```
gifsicle -O3 --lossy=80 --colors 256 plane.gif -o plane_giflossy.gif
```

これは絶対に驚くべきバイトサイズ`212666`を持っています。注文を使用して達成したものほど高品質ではありません ディザリングしますが、サイズは1/2未満です。

上記の結果に勇気づけられて、私はGifLossyを最良の順序のディザで使用することにしました それがそれをさらに小さくすることができるかどうかを確認するために、私たちが得た結果。

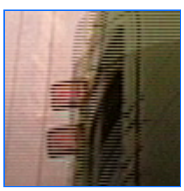
```
gifsicle -O3 --lossy=80 plane_od2.gif -o plane_od2_giflossy.gif
```

そして、バイトのサイズ`202881`はさらに小さくなりました。残念ながら、基本的に高品質の注文ディザを失いました 私たちが以前に達成するために一生懸命働いている結果。これは残念です。

ビデオフレームのインターレース解除

すべての画像がデジタルカメラからのものであるとは限りません。画像を抽出することは非常に一般的です 非CCDビデオカメラからのデジタルビデオフィードから。これらの画像は テレビに直接表示するためにインターレースされ、2行ごとに 画像の異なるフレーム(インターレース)。物事が動いていない2つのフレームの場合、インターレースは通常あまりありません 顕著。おそらく、画像のわずかなエッジのぼやけを生成します。しかし、いつ 動きの速いオブジェクトが関与している場合、結果として得られるインターレース画像は非常に 2つのフレームがマージされているため、当惑します。

ヴォルフガング・ヒューゲマン・<Auto@Hugemann.de>(ドイツ)は、この問題を抱えていて、クラッシュテストのスナップショットを送ってくれました。ヴォルフガングは自分を連れて行った。しかし、デモンストレーションのために私はより小さな画像を使用します これからトリミングされました。ただし、テクニックはフルサイズで機能します 画像。



```
magick video_frame.png  -crop 100x100+200+470 +repage  interlaced.png
```

ヴォルフガング・ヒューゲマンは、元のビデオフレームにTIFF形式を使用しました。IMの例で使用するためにこれをPNGに変換しました。使用しようとしないでください これらの画像のJPEGは、処理が完了するまで このプロセスに必要な低レベルの品質を破壊します。

ご覧のとおり、インターレースは2つの別々のフレームを示しています。インターレースPALデジタルビデオシーケンス(約50ハーフフレーム/秒)。はい、車は非常に速く動いていて、カメラは高速シャッターを使用しています、非常に高品質のビデオ画像を生

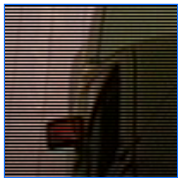
成します。結果の画像は2つです 車のサイドミラーがかなりの距離を移動する織り交ぜられたハーフフレーム ハーフフレーム間の間の1/50秒の期間中。

ここでは、インターレースされたハーフフレームの1つ(2行ごと)を置き換えます。 白と。これは、「BoB」として知られる標準的なインターレース解除方法です。 フィルター。これは、IMの例のためにヴォルフガングによって寄稿されました。

```
magick interlaced.png  -fx "floor(j/2)==j/2 ? u : 1"  deinterlace_1.png
```

FX演算子が遅いので、代替とは、「縞模様の画像」を作成することです。そのような画像を生成することができます 特別な「」組み込みイメージから。pattern:Horizontal2

その画像は、''合成方法を使用してオーバーレイして、元の画像とオーバーレイできます 白い線を使用するか、''を使用するか、黒い線をオーバーレイします。例えば。。。ScreenMultiply



```
magick -size 100x100 pattern:Horizontal2 \
        interlaced.png -compose Multiply -composite  deinterlace_2.png
```

パターンの否定を使用して、インターレースの残りの半分を選択できます 画像。または、''を ''に変更した場合 背景が白のフレームを抽出できます。MultiplyScreen

別の方法として、不足しているフレームラインを埋めようとしました 前の行を複製します。

```
magick interlaced.png  -fx "u.p{i,j-j%2}"  deinterlace_3.png
```

ピクセル化を使用することもできます 毎秒倍増するように画像を縮小・拡大する手法 線。

```
magick interlaced.png -sample 100%x50% \
                      -sample 100%x200%  deinterlace_4.png
```

そして、わずかなバリエーションで、両側の線を組み合わせて サイズ変更拡張の一環として、ハーフフレーム画像を垂直方向に滑らかにします。

```
magick interlaced.png -sample 100%x50% \
                      -resize 100%x200%  deinterlace_5.png
```

その結果、インターレースの1つのフレームが特にうまく抽出されます。ビデオ画像。

画像から残りのハーフフレームを抽出する場合は、調整できます '(IM v6.8.4-7 以降)sampling:offset

```
magick interlaced.png -define sample:offset=75 \
        -sample 100%x50%  -resize 100%x200%    deinterlace_6.png
```

このバージョンのIMより前は、画像を1ピクセルずつ「」する必要があります。同じ結果を達成します。-roll

作成日:2007
年2月28日 更新日: 2010年10
月7日 著者:アンソニーティッセン、<Anthony.Thyssen@gmail.com>
例で生成:
URL: https://imagemagick.org/Usage/video/