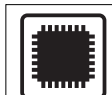


続

実設計に応用できる 演算回路スキルを 身につけよう



デバイスの記事

外村元伸

第1回 デジタル画像処理のための各種補間法を理解する

本連載では、デジタル画像処理機器で利用される画像の拡大および縮小スケーリングを題材に、高性能な演算回路の設計方法を解説する。今回は、デジタル画像処理で用いられる「アルファ・ブレンディング」や画像拡大・縮小スケーリングのための各種補間法の基礎を取り上げる。なお、本連載は本誌2006年5月号、pp.39-48、「特集1 第2章 LSIの中の基本部品を原理から理解する」で紹介した2進数や演算器の基本回路構成、線形補間の考えかたをベースにして話を進める。

(編集部)

本誌2006年5月号、pp.39-48、「LSIの中の基本部品を原理から理解する」では、線形補間という身の回りの製品でも利用されている演算の専用回路化を題材に、演算器とその回路設計の基礎について解説しました。本連載は、その具体的な応用例として、「画像処理の演算器設計」を取り上げます。画像処理を選んだ理由は、表示された画像を見れば自分の作った回路のよしあしが一目瞭然だからです。また、最近はいろいろな分野でデジタル画像処理機器が普及しています。みなさんの中にもこれらの設計に携わっている、あるいは今後携わっていく方も少なくないでしょう。本稿がみなさんのおしごと役に役立つことを期待しています。

1. 画像合成「アルファ・ブレンディング」

二つの画面AとBを割合 α ($0 \leq \alpha \leq 1$)で合成表示することを考えた場合、これは線形補間です。割合 α にちなんで、画面合成という行為は「アルファ・ブレンディング」と

呼ばれています。ここでは、アルファ・ブレンディングの画面合成に線形補間演算器を応用した特徴的な実現法を紹介します。

● 循環2進小数なので簡単化できる

デジタル・テレビの規格では、合成の割合 α は8ビットの整数(10進で0~255)で定義しています。これが α レジスタに格納されているとします。 α レジスタの内容が0と255のとき、それぞれAとBの一方の画面のみを表示するようになっています。

α レジスタの内容から0~1の表現に変換するためには、図1に示すように $1/255 \times \alpha$ レジスタの計算を行わなければなりません。普通に考えれば、これには乗算が必要になります。

実際には、 $1/255$ の定数乗算を行うのを避けて、近似的に $1/256$ として α レジスタの内容の単純な固定小数点位置のシフトに置き換えています。そして、比較器で α レジスタを判定し、レジスタの内容が255であれば画面Bのみを表示するように設計しているようです。

ところで2006年5月号、pp.39-48の記事の中で、循環2進小数の例として、

$$1/255 = [0.0000000100000001...]_b = [0.00000001]_b$$

を挙げました。 α レジスタ $= [d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0]_b$ とすれば、 $1/255 \times \alpha$ レジスタ $= [.d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0]_b$ と単純に表現できるので、適当なビット位置 i で $d_i = 1$ ならば切り上げるだけでよいのです。任

KeyWord

演算回路、画像合成、アルファ・ブレンディング、線形補間、最近傍選択法、バイリニア法、バイキュービック補間法、コンボリューション法、スプライン法

意のビット長 n の場合も次の式 (1) が成り立つので、循環 2 進小数になっています。

$$\frac{1}{\sum_{i=0}^{n-1} 2^i} = \frac{1}{2^n - 1} = \sum_{j=1}^{\infty} 2^{-jn} = 2^{-n} + 2^{-2n} + \dots \quad \dots\dots\dots (1)$$

2. 画像拡大・縮小スケーリング法いろいろ

次に、画像の拡大・縮小スケーリング表示へ線形補間演算を応用してみましょう。実際の応用場面では、線形 (1 次) 補間では不十分で、2 次補間、3 次補間が必要な場合が少なからずあります。ここでは、3 乗演算を含む 3 次補間演算器の設計法について詳しく説明します。こう言うと、「3 乗を含む演算は複雑なためコストがかかる。直接演算はあきらめて、演算結果のテーブルを利用して設計しよう」とたいていの方は考えるのではないのでしょうか。

しかし、線形補間演算器の原理や基礎を押さえていれば、ちょっとしたくふうで意外と簡単に、しかもコンパクトに直接演算を用いて実現できてしまいます。従来のみなさんの常識を変えたいというのが本稿のねらいです。またここで解説するテクニックは、画像処理以外でも、例えば音声処理や通信処理など、いろいろな分野に応用できます。

● 表示画面の違う機器でデータを共有するとき必要な技術

ひと口にデジタル画像処理と言っても、民生機器 (テレビ、パソコン、ゲーム機、カメラ、プリンタなど)、携帯

電話、車載機器、医療機器など、応用分野はさまざまです。

これらの機器には、それぞれ入力用のカメラと出力用のディスプレイが必要に応じて付いています。また、業務用と個人用、机上用と携帯用、大型と小型、動画用と静止画用、高解像度と低解像度などという分類もあります。これら各種デジタル画像処理機器で取得した画像データを互いに転送して共用しようとするとき、機器の表示画面に応じた画像の拡大・縮小変換が要求されます。これを画像の拡大・縮小スケーリングと呼んでいます。

この場合、適当な原画像の補間計算によって表示位置の画素値を求める必要があります。とくに、つぶれないように縮小すること、滑らかに拡大することが要求されるため、提案されている各種補間方式の中から、画質に応じて適切な方式を選択しなければなりません。

● 直接演算すればコンパクトで高速なスケーリングが可能

さまざまな状況の画質で画像が扱われるため、画像の拡大・縮小スケーリングのための補間計算法の種類が多様化し、複雑化してきています。それらに伴う回路規模や消費電力の増大はできるかぎり抑えなければなりません。

現状ではこうした拡大・縮小スケーリングについて最適な演算器設計がなされているとは思えません。3 次多項式の計算は複雑だという考えから、規模が大きいにもかかわらず演算結果のテーブルを用いて実現しているからです。テーブルというメモリを使うことで、メモリの参照時間が余計にかかります。これを直接演算することにより処理時

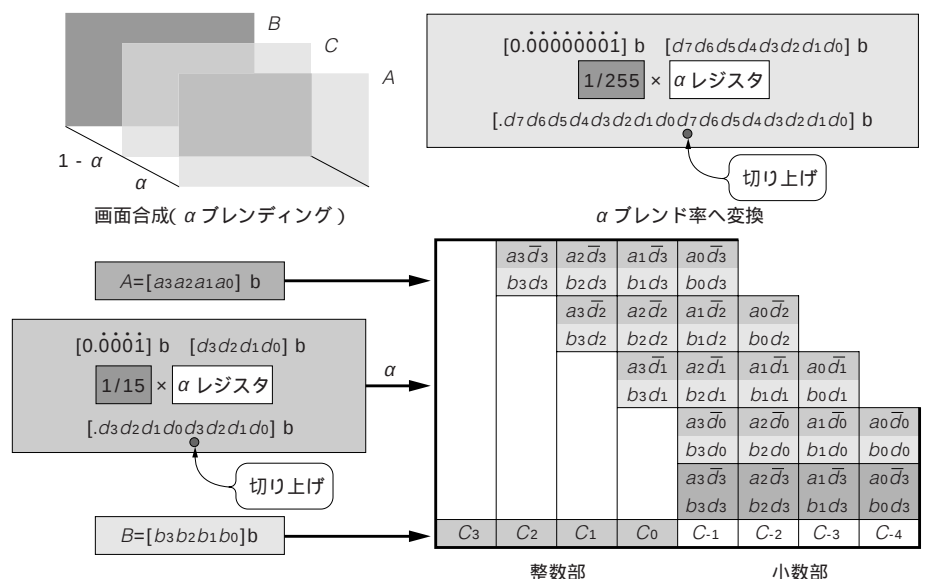
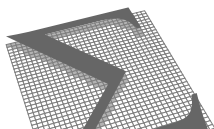


図 1
アルファ・ブレンド画面合成へ線形補間演算器を応用
積項の積み上げ表現形式については、誌面のついで 4 ビットの例 (1/15) × (α レジスタ) を示す。d₃ による切り上げの項が余分に加わる

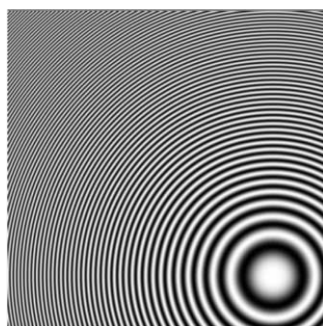


間を短縮できます。

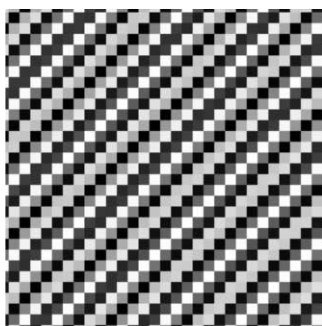
また、直接演算することで、演算精度も容易に保証できます。そのため、演算幅のサイズを最小化でき、演算器をコンパクトにすることができます。

● 補間精度を上げるにはバイキュービック補間法を用いる

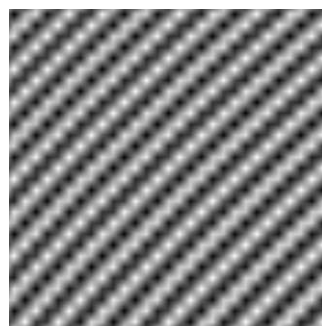
画像の拡大・縮小スケーリングによく使われるのは、最近傍選択(nearest-neighbor)法とバイリニア(bilinear)補間法です。



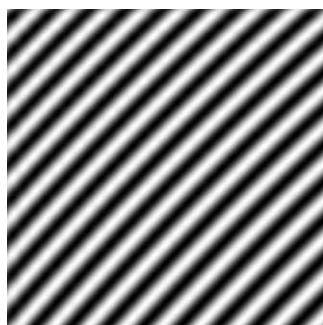
(a) 原画像



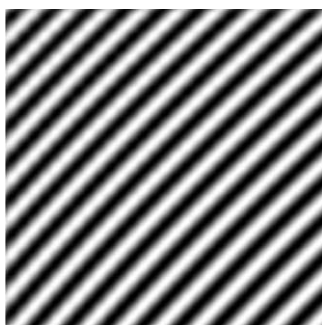
(b) 最近傍選択



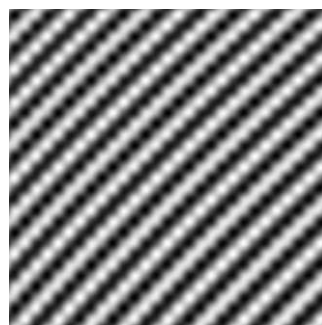
(c) バイリニア



(d) バイキュービック・スプライン6点



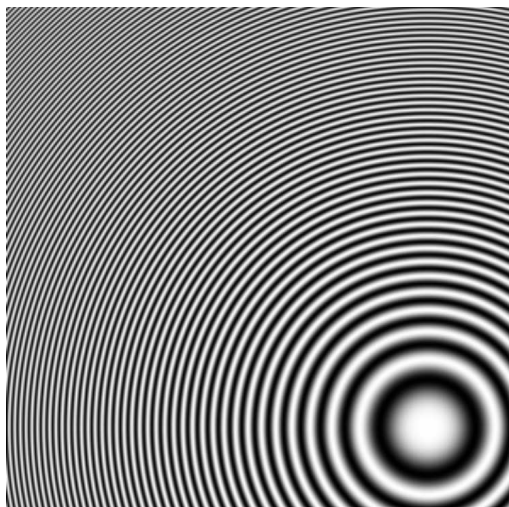
(e) バイキュービック・スプライン4点



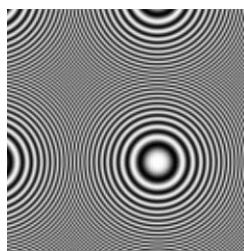
(f) バイキュービック・コンボリューション

図2 各種補間法に基づく画像拡大スケーリング表示結果の比較

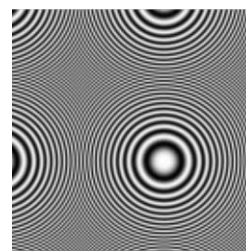
各辺の拡大率が10倍(面積拡大率100倍)の場合の表示比較を示す



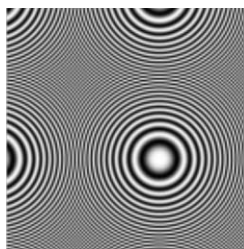
(a) 原画像



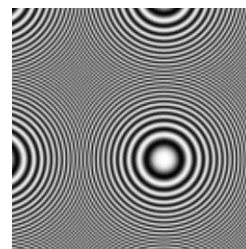
(b) 最近傍選択



(c) バイリニア



(d) バイキュービック・スプライン4点



(e) バイキュービック・コンボリューション

図3 各種補間法に基づく画像縮小スケーリング表示結果の比較

各辺の縮小率が1/3倍(面積縮小率1/9倍)の場合の表示比較を示す。拡大の場合と比べて補間方式による違いがはっきりとわからない