

2015 年度 卒業研究論文

画像補間法による拡大

岡山理科大学

総合情報学部 情報科学科

I 1 2 I 0 6 1 田中 宏美

I 1 2 I 0 6 3 仲 陽美

目次

1	はじめに.....	1
2	ラスタ画像.....	1
2.1	bit数による差.....	2
3	画像拡大と画像補間法.....	2
3.1	ニアレストネイバー法.....	2
3.2	バイリニア法.....	4
3.3	バイキュービック法.....	6
3.4	Lanczos(n)法.....	7
4	拡大画像の比較と評価.....	10
4.1	画像補間法を用いた拡大画像の比較と評価.....	11
4.2	グレイスケール画像とカラー画像の比較と評価.....	12
4.3	8bitグレイスケール画像と16bitグレイスケール画像の比較と評価.....	17
4.4	24bitカラー画像と42bitカラー画像の比較と評価.....	17
5	Lanczos(n)法の違い.....	18
6	まとめ.....	19
	参考文献.....	19

1 はじめに

近年，スマートフォンやデジタルカメラの普及により，デジタル画像はより身近なものになっている．パソコンの画面やデジタル写真に使われているデジタル画像は一般に1画素あたり24bitのデータを有している．しかし，デジタル画像には24bit以外のbit数を有する多くの種類の画像が存在する．

本研究では，デジタル画像のデータ長の差異に着目している．bit数の異なる画像を，画像補間法を用いて拡大し，出力画像の特徴などについて比較をする．多項式を用いた画像補間法には，画像を構成する特定の画素の近傍を利用する方法として代表的な，ニアレストネイバー法，バイリニア法，バイキュービック法，Lanczos(n)法が存在する．画像補間法の特徴を比較したのちに，画像拡大をした場合の出力画像の違いを評価し，使用場面についてまとめる．

2 ラスタ画像

ラスタ画像とは，一般的に使用されている画像のことであり，色のついた小さな点（ドット）を2次元配列として表現したデータのことである．標準画像データベースにより配布されているParrotsも，ラスタ画像のひとつである（図1）．複雑な図形や色の組み合わせられたデータ，例えば写真などを扱うにはこのラスタ画像が適している．ある画像を構成する点の最小の単位を画素（ピクセル）と呼ぶ．例えば1画素24bit画像で 256×256 画素の画像は，横方向256個，縦方向256個で合計65536個の画素で構成されている．カラー画像の各画素にはRGB（Red, Green, Blue）すなわち光の3原色に対応する画素値が各8bitを用いて与えられており，計 $256 \times 256 \times 256$ 色を表すことができる．Webブラウザが対応している主な画像形式としてはラスタ画像が採用されており，OSに付属するような簡易グラフィックス処理ソフトはラスタ画像のみ扱うようになっている．ラスタ画像の代表的な規格には，BMP, GIF, JPEG, PICT, PNG, TIFFなどがある[1]．



図1 ラスタ画像（標準画像データベース Parrots）と拡大画像

2.1 bit 数による差

デジタル画像は用いる bit 数によって表現できる色の数が変わる。画素を構成する原色 RGB それぞれに画素値が存在している。カラー画像は RGB それぞれの色の階調（すなわち、色や明るさの濃淡の段階数）の組み合わせによって色が表現される。24bit カラー画像の場合は、赤に対して 8bit、緑に対して 8bit、青に対して 8bit の合計 24bit の画素を有している。8bit は 256 階調になるため、24bit カラー画像は $256 \times 256 \times 256$ 色を用いて表現する（図 2）。48bit カラー画像（ただし、本研究で使用する画像は RGB14bit の RAW データ画像になり、情報量は 42bit である）の場合は、形式的ではあるが赤、青、緑のそれぞれに対して 16bit 合計 48bit を用いているので、 $65536 \times 65536 \times 65536$ 色を表現することができる（図 3）。



図 2 原画像 (24bit カラー画像)



図 3 原画像 (42bit カラー画像)

3 画像拡大と画像補間法

画像補間法とは、画面の拡大、縮小、回転、変形、解像度の変更などの操作を行うとき、元の画素値を用いて必要とされる新規の画素値を求め、近傍にある画素値を参照して補間計算する方法である。近傍を利用する代表的な画像補間法にはニアレストネイバー法、バイリニア法、バイキュービック法の 3 手法がある。また近年、Lanczos(n)法を活用した画像補間法も画像編集ソフトに採用され、注目されている[2], [3]。

3.1 ニアレストネイバー法

ニアレストネイバー法は、0 次多項式を用いる画像補間法である。最近傍にある画素値を単純にコピーし新規の画素値として使用する方法である（図 4）。すなわち、原画像をもとにそのまま代入処理だけを介して拡大する方法である。また、先に挙げた 4 通りの補間法の中でもっとも処理速度が速く、色相が変化しない画像補間法である。そのため、単純にそのまま拡大したい場合に用いるのが適している。しかし、この方法は画質が粗くなる傾向にあり、画像中の対象物の輪郭にジャギーが目立ちやすく階調の変化もあるような画像の拡大にはむいていない。ジャギーとは、文字や画像を縮小・拡大表示すると目立つ階段状のギザギザのことである。

補間する際の求めたい新規の画素の座標 $(x + d_x, y + d_y)$ の画素値 I には原画像の最近接にある画素値 f を用いる。原画像の画素値との距離を d ，原画像の画素値の重みを $W_0(d_x, d_y) = w_0(d_x)w_0(d_y)$ として計算した場合の式は以下の通りである（図 5）。重み $W_0(d_x, d_y)$ から原画像の最近接点を選び画素値をコピーし，求めたい座標の画素値を定め，画像の拡大を行う。座標変位 d_x, d_y は画素値を補間したい座標を表し， d の値は求めたい座標と参照する座標との距離を示しているため補間処理を進めていくにしたがって変化する（以下の 3 通りの補間法でも同じである）。

$$0 \leq d_x \leq 1, 0 \leq d_y \leq 1$$

$$w_0(d) = \begin{cases} 1 & (|d| < 0.5) \\ 0 & (0.5 < |d|) \end{cases} \dots\dots\dots (式 1)$$

$$I(x + d_x, y + d_y) = f(x, y)W_0(d_x, d_y) + f(x + 1, y)W_0(1 - d_x, d_y) + f(x, y + 1)W_0(d_x, 1 - d_y) + f(x + 1, y + 1)W_0(1 - d_x, 1 - d_y) \dots\dots\dots (式 2)$$

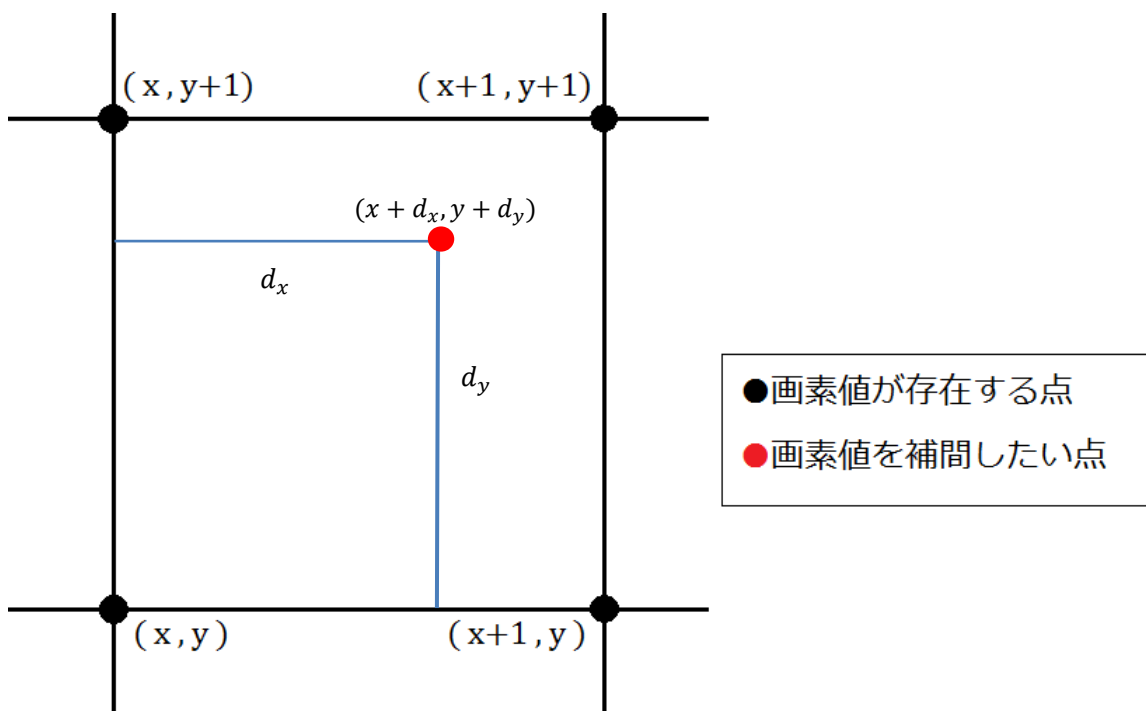


図 4 ニアレストネイバー法

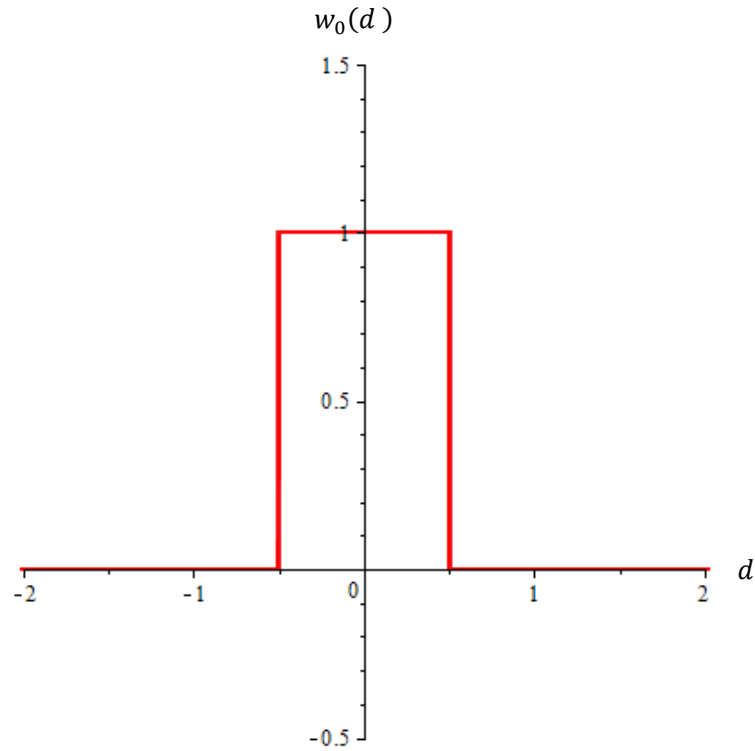


図5 ニアレストネイバー法の重み

3.2 バイリニア法

バイリニア法は、1次多項式を用いる画像補間法である。2×2画素単位で画素値を補間する方法である。バイリニア法では補間される画素のまわりの隣接する4点から線形型計算により画素値を求めている（図6）。これもニアレストネイバー法と同様に計算処理の少ないことから補間にかかる時間は短い。画像の粗さが目につく。ニアレストネイバー法は原画像にある色のみで補間するため、ジャギーが目立ち画像の粗がはっきり見えてしまう。バイリニア法は参照する画素値の中間の色を生成することでジャギーが目立たず、ニアレストネイバー法を用いて拡大した画像よりもきれいに見える。中間の色を生成するため、輝度の変化が激しいピクセル境界部分では、ぼやけて表示されるので画質はよくない。

補間する際の求めたい座標 (d_x, d_y) の画素値 I には原画像の画素値 f を用いる。座標 (d_x, d_y) と原画像の画素値との距離を d 、原画像の画素値の重みを $W_1(d_x, d_y) = w_1(d_x)w_1(d_y)$ として計算した場合の式は以下の通りである（図7）。ただし、距離 d は参照する座標が4点あるため、参照する座標に対応しているものとする。

$$0 \leq d_x \leq 1, 0 \leq d_y \leq 1$$

$$w_1(d) = \begin{cases} 1 - |d| & (|d| < 1) \\ 0 & (1 < |d|) \end{cases} \quad \dots \dots \dots \text{(式3)}$$

$$I(x + d_x, y + d_y) = f(x, y) W_1(d_x, d_y) + f(x, y + 1) W_1(d_x, 1 - d_y) \\ + f(x + 1, y + 1) W_1(1 - d_x, 1 - d_y) + f(x + 1, y) W_1(1 - d_x, d_y) \\ \dots \dots \dots \text{(式4)}$$

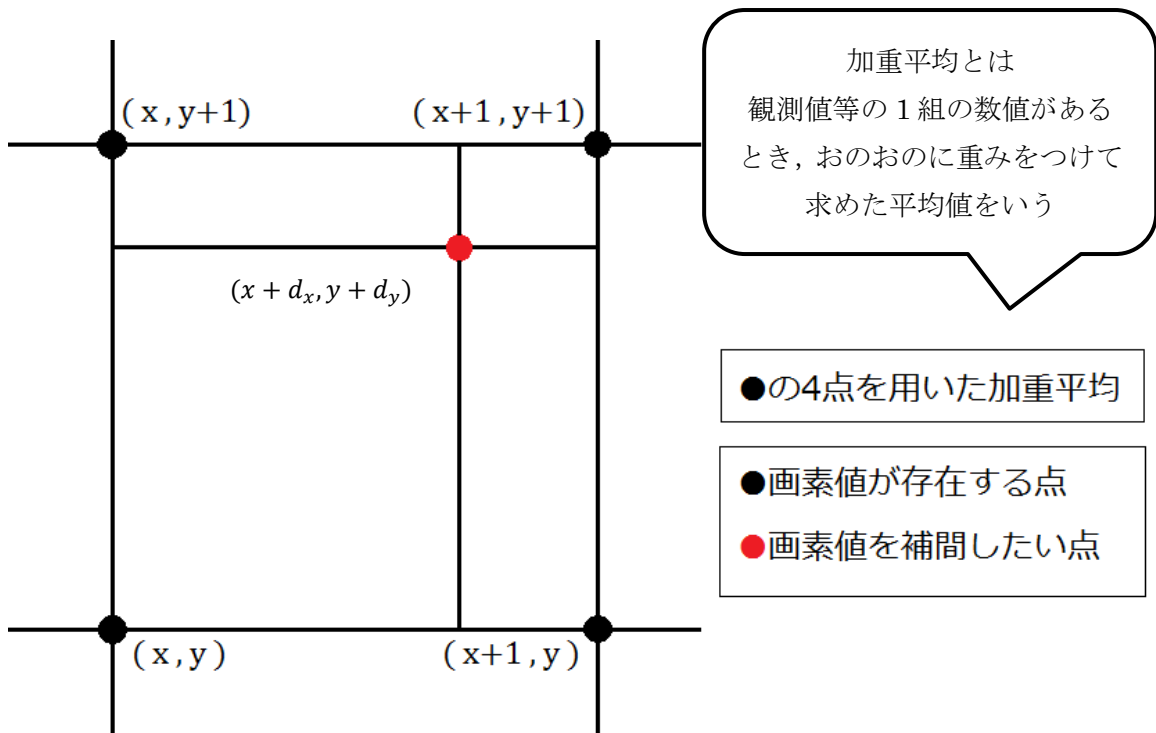


図6 バイリニア法

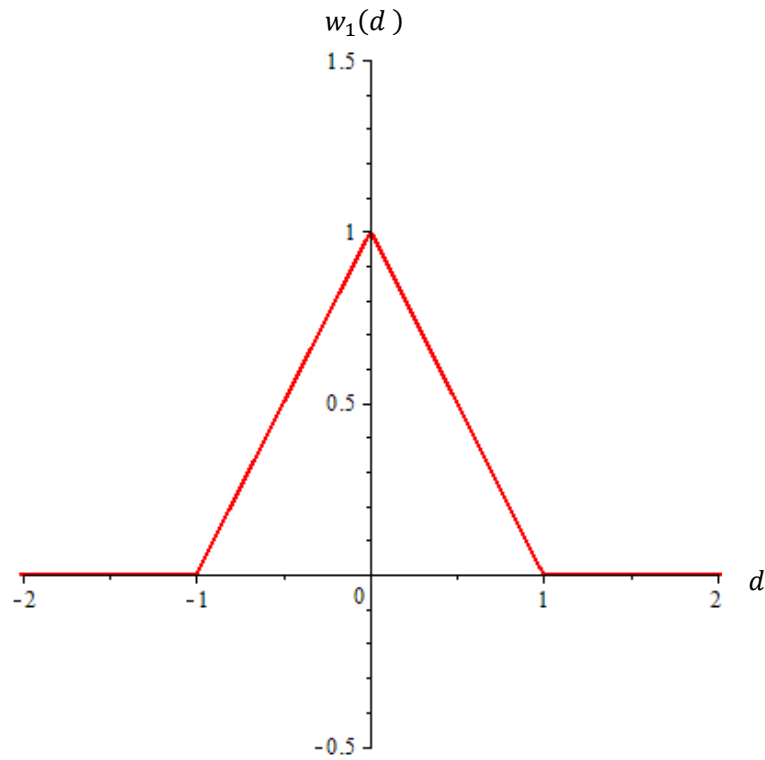


図7 バイリニア法の重み

3.3 バイキュービック法

バイキュービック法は、3次多項式を用いる画像補間法である。すなわち、4×4単位で画素を補間計算する。補間される画素のまわりの16点を用いて3次多項式を用いた近似補間を行い、画素値を決める方法である(図8)。代表的な4通りの画像補間法の中で、元の画像に含まれている情報の損失が最も少なく、自然な画像が得られるもっとも精度の高い画像補間法であることが知られている。しかし、上記の手法よりも計算量の多い演算を行なうため処理に時間がかかる。バイリニア法の約2倍の計算処理を行っている。

補間する際の求めたい座標 (x, y) の画素値には原画像の画素値を用いる。座標 (x, y) と原画像の画素値との距離を d 、鮮鋭度を a 、原画像の画素値の重みを $W_2(d_x, d_y) = w_2(d_x)w_2(d_y)$ として計算した場合の式は以下の通りである。画像編集ソフト内で使用される鮮鋭度 a は-0.5や-1.0などを代入して用いている。本研究では、 $a = -1.0$ としてsinc関数と対応する3次多項式を用いている(図9)。ただし、距離 d は参照する座標が16点あるため、 d_1 から d_{16} と示し、参照する座標に対応しているものとする。 d_1 は $f(x_1, y_1)$ との距離でここを中心として考える。他の座標についても同様にして d の座標を決定していく。

$$w_2(d) = \begin{cases} (a+2)|d|^3 - (a+3)|d|^2 + 1 & (|d| < 1) \\ a|d|^3 - 5a|d|^2 + 8a|d| - 4a & (1 < |d| < 2) \\ 0 & (2 < |d|) \end{cases} \dots\dots (式5)$$

$$I(x, y) = f(x_0, y_0)W_2(1+d_x, 1+d_y) + f(x_1, y_0)W_2(d_x, 1+d_y) + \dots + f(x_2, y_3)W_2(1-d_x, 2-d_y) + f(x_3, y_3)W_2(2-d_x, 2-d_y) \dots\dots (式6)$$

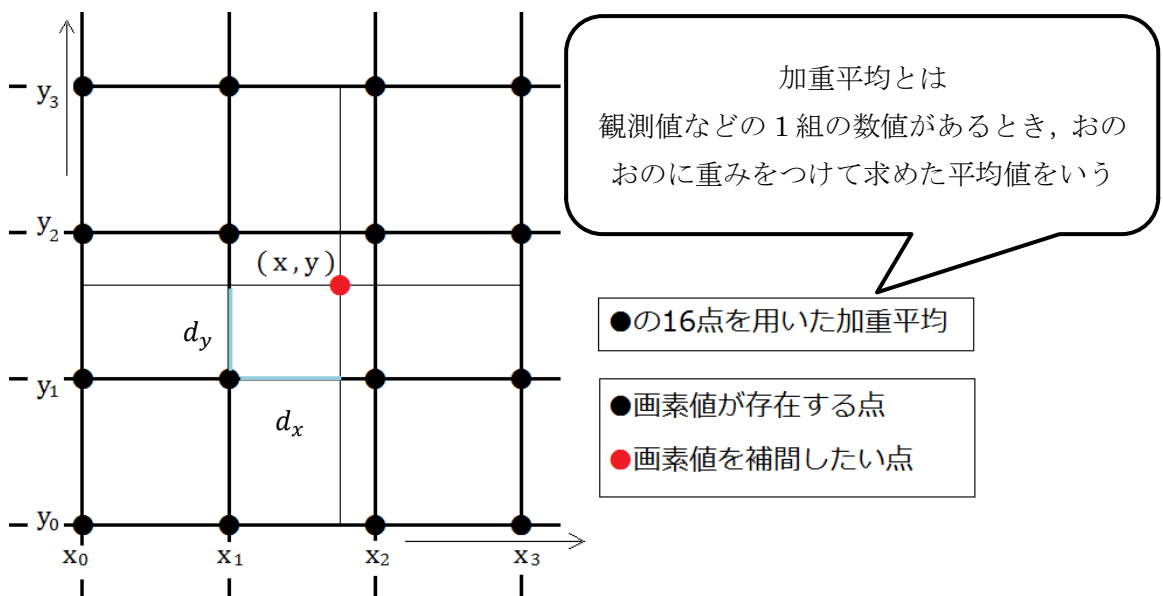


図8 バイキュービック法

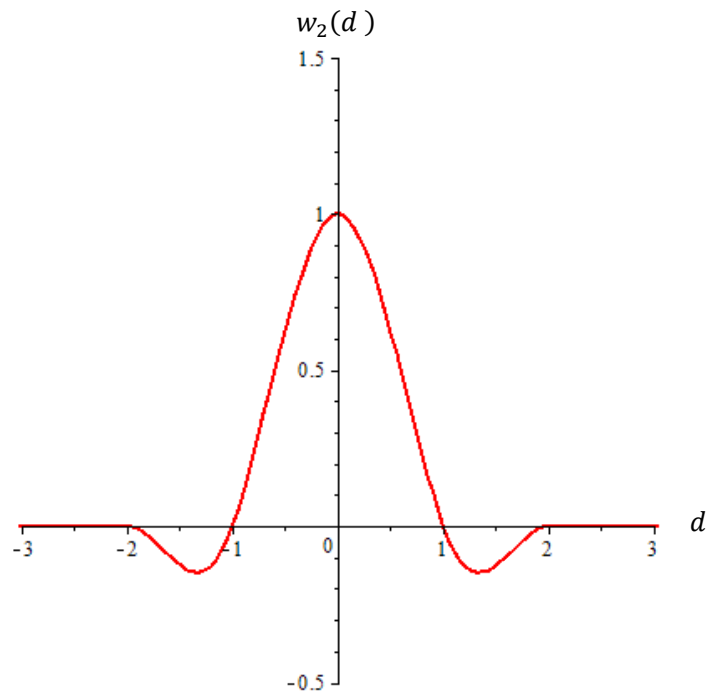


図 9 バイキュービック法の重み

3.4 Lanczos(n)法

Lanczos(n)法は、sinc 関数を用いた画像補間法である (式 7)。バイキュービック法は補間される画素のまわりの 16 点のみを用いるが、この手法では参照する画素の数を簡単に調整することができるため 16 点以上の画素を用いることもある (図 11)。その反面、上記 3 手法よりも計算量の多いことから、処理時間は長くなる。多くの画素を参照できるため、バイキュービック法と同等以上の高画質な結果を出すことができるといわれている。例えば、Lanczos(2)では 4×4 単位の画素、Lanczos(3)では 6×6 単位の画素を参照している。すなわち、Lanczos(n)では $2n \times 2n$ の画素を参照している。画像編集ソフトで実装されているものの多くは Lanczos(3)や Lanczos(4)などである [3]。

補間する際の求めたい座標 (x, y) の画素値には原画像の画素値を用いる。座標 (x, y) と原画像の画素値との距離を x 、原画像の画素値の重みを $W(x, y)$ として計算した場合の式は以下の通りである。式中の n に任意の数を入れることによって参照する画素の数を変更することができる。以降では、 $n = 4$ として扱っている。

$$\text{sinc}(x) = \sin(\pi x) / \pi x \quad \dots \dots \text{(式 7)}$$

$$W(x, y) = w(x)w(y)$$

$$w(x) = \begin{cases} \text{sinc}(x) \text{sinc}\left(\frac{x}{n}\right) & (|x| < n) \\ 0 & (n < |x|) \end{cases} \quad \dots \dots \text{(式 8)}$$

したがって補間を行うのは x 座標 p から Δp 離れた場所とすると、その場所への補間に用いる画素の座標を q とした場合、補間に用いる画素の座標と補間を行う場所との距離が $p - q + \Delta p$ なので、重み関数 $W(p - q + \Delta p)$ は以下のようになる (図 10)。

$$W(p - q + \Delta p) = \begin{cases} \text{sinc}(p - q + \Delta p) \text{sinc}((p - q + \Delta p)/n) & (|p - q + \Delta p| < n) \\ 0 & (n < |p - q + \Delta p|) \end{cases} \dots (\text{式 } 9)$$

上記の重み関数を使った処理を先に示したように、横(x)方向、そして縦(y)方向に適応して拡大(補間)画像を得ることができる(図11)。Lanczos(4)法の重みグラフからバイキュービック法よりも多くの画素を参照していることがわかる(図12)。この処理をCプログラムで実行する(図13)。このプログラムでは、簡単のため、倍率NNを整数にしている。

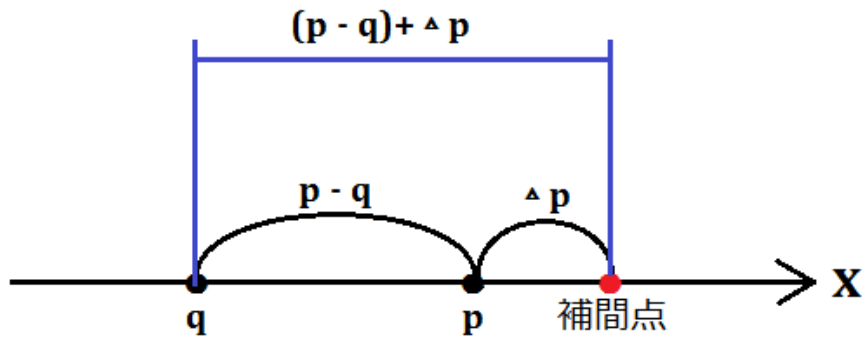


図10 重み関数 $W(p-q+\Delta p)$ と座標の関係

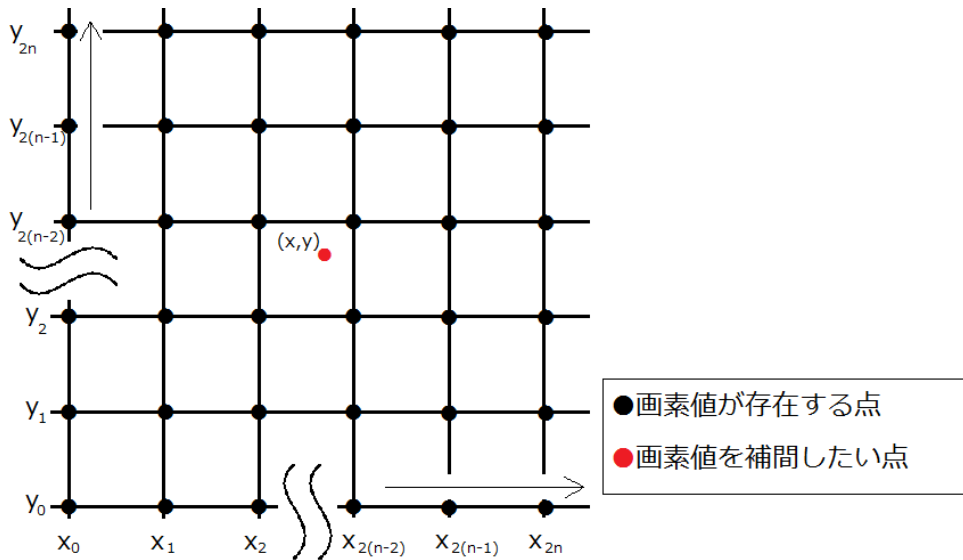


図11 Lanczos(n)法

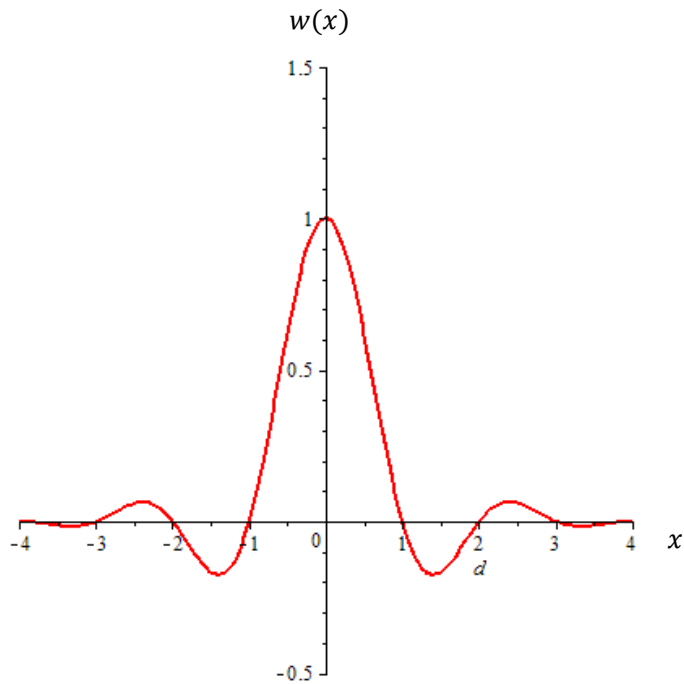


図 12 Lanczos(4)法の重み

```

int sincLn=4, NN=4;          //NN は倍率
//カーネルの計算
for(row=0; row<NN-1; row++){
  for(col=0; col<sincLn*2; col++){
    SL[row][col]=sin( dblPi*( (row+1.)/NN -col-1+sincLn ) )
                  *sin( dblPi*( (row+1.)/NN -col-1+sincLn )/sincLn )
                  /( (dblPi*( (row+1.)/NN -col-1+sincLn ) )
                    * ( dblPi*( (row+1.)/NN -col-1+sincLn )/sincLn ) ); }}
//原画像のコピー
for(y=0; y<height; y++){
  for(x=0; x<width; x++){
    imageOrg.at<Vec3b>(y,x)=imageInput.at<Vec3b>(y,x);
    imageBig.at<Vec3b>(y*NN,x*NN)=imageOrg.at<Vec3b>(y,x);  }}
//原画像を x 方向に拡大(画素 NN-1 個を計算)
for(y=0; y<height; y++){
  for(x=sincLn-1; x<width*sincLn; x++){
    for(n=0; n<NN-1; n++){
      //m=0,1,2 ... BGR channel color data => imageBigm (同じ構造にしてから計算)
      for(m=0; m<3; m++){
        colorTmp=0.0 ;

```

```

    for(col=0; col<sincLn*2; col++){
        imageBigm=imageBig.at<Vec3b>(y*NN,(x+col+1-sincLn)*NN)[m];
        colorTmp+=SL[n][col]*imageBigm; }
        colorTmp=MIN(65535,MAX(colorTmp,0));
        imageBig.at<Vec3b>(y*NN,x*NN+n+1)[m]=colorTmp; }}
    }}
for(x=(sincLn-1)*NN; x<(width-sincLn)*NN; x++){
    //原画像を y 方向に拡大(画素 NN-1 個を計算)
    for(y=sincLn-1; y<height-sincLn; y++) {
        for(n=0; n<NN-1; n++){
            //m=0,1,2 ... BGR channel color data => imageBigm (同じ構造にしてから計算)
            for(m=0 ; m<3; m++){
                colorTmp=0.0 ;
                for(col=0; col<sincLn*2; col++){
                    imageBigm=imageBig.at<Vec3b>((y+col+1-sincLn)*NN, x)[m];
                    colorTmp+=SL[n][col]*imageBigm; }
                    colorTmp=MIN(65535,MAX(colorTmp,0));
                    imageBig.at<Vec3b>(y*NN+n+1,x)[m]=colorTmp; }}
            }}
    }}

```

図 13 Lanczos(4)法の計算プログラム

プログラム中の変数 `sincLn` により Lanczos(n)法の n の値を決めている。 `SL[row][col]`は計算負荷を少なくするよう $W(p - q + \Delta p)$ を表にしたものである。ここでは計算量の増えないうよう、まず横方向に拡大し、その次に縦方向に拡大している。

4 拡大画像の比較と評価

画像補間法で拡大した様々な画像の比較を行い主観評価と客観評価を行う。画像の評価方法として、目視による主観評価と画素値を基に比較用の評価値を計算する客観評価がある。画像の評価方法として主に使われているのは主観評価である。本研究では bit 数による差を目視で評価をしたのち、再現性を確認するための数値を用いた客観評価も行うことにした。本研究ではこの客観評価にエントロピ (entropy) を使用している。エントロピとは画像の平均情報量を表す指標で単位は bit である。エントロピが大きいほど画像の有している情報が多いことになる。以下では、拡大した画像を画像補間法の名前で区別して示している。

4.1 画像補間法を用いた拡大画像の比較と評価

それぞれの画像補間法の比較を行う。原画像は 24bit カラー画像 Parrots (図 14) を用い、図中の赤枠で囲まれた部分を比較が容易となるよう拡大している。

主観評価によると、ニアレストネイバー法 (図 15(a)) では、ジャギーが目立つことが確認できる。バイリニア法 (図 15(b)) では、ニアレストネイバー法に比べジャギーは改善されているが、輪郭がぼやけて見えることが確認できる。バイキュービック法 (図 15(c)) では、ニアレストネイバー法とバイリニア法の比較しジャギーがより改善され輪郭も少しははっきりしていることが確認できる。Lanczos(4)法では、バイキュービック法と比べると輪郭がよりはっきりしていることが確認できる。客観評価に用いたエントロピによる比較 (表 1) では原画像 (図 14) とニアレストネイバー法 (図 15(a)) は同じ値になった。それ以外の画像補間法ではバイリニア法 (図 15(b))、バイキュービック法 (図 15(c))、Lanczos(4)法 (図 15(d)) の順に少しずつ増えているのがわかる。このことからニアレストネイバー法は原画像と同じ情報量であること、それ以外の方法は原画像の画素値のほかに新たに多数の画素値を補間しているため、原画像と比べると情報量の増えていることが確認できる。

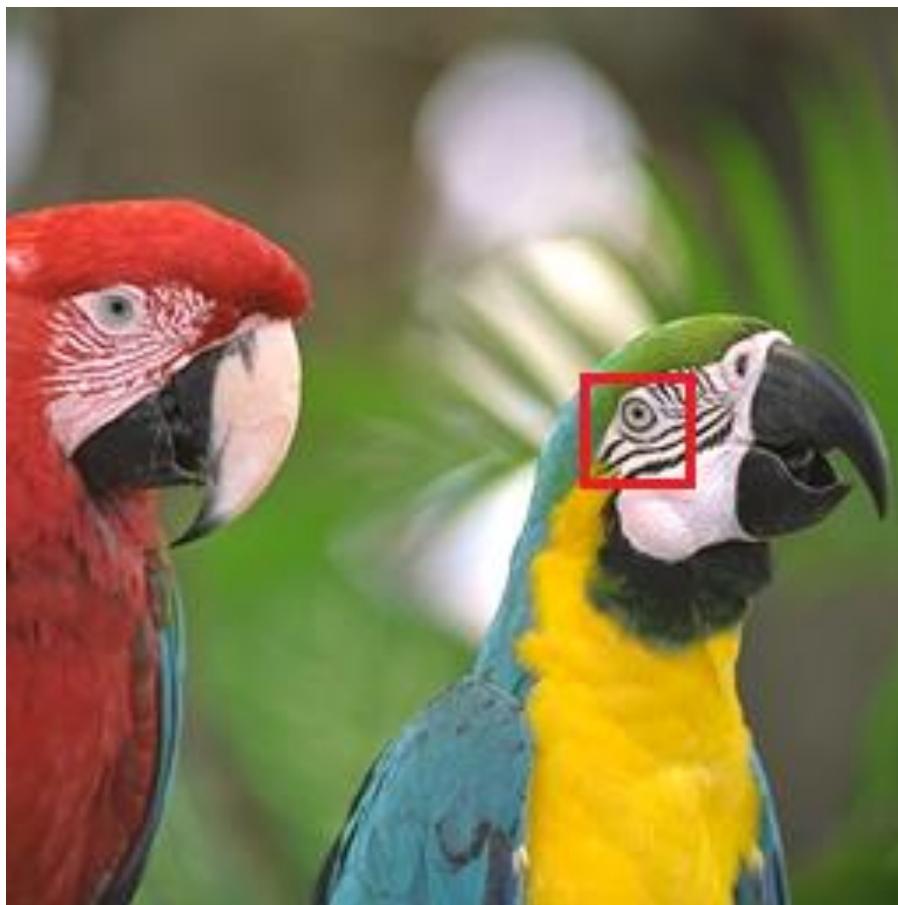
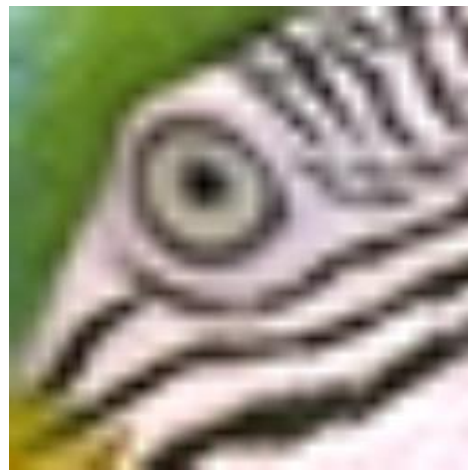


図 14 原画像 (Parrots)



(a)ニアレストネイバー法



(b)バイリニア法



(c)バイキュービック法



(d)Lanczos(4)法

図 15 画像 Parrots の拡大画像

表 1 Parrots のエントロピ

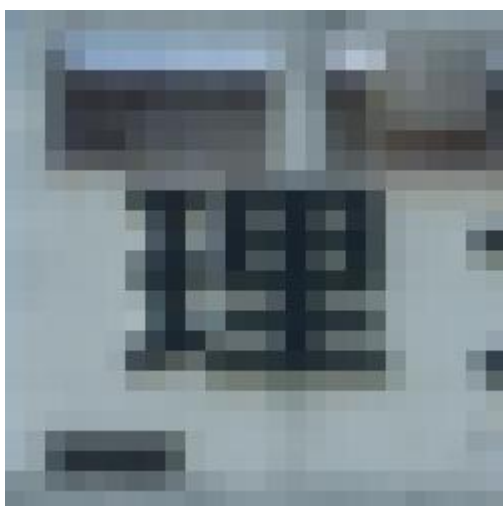
原画像 (図 13)	7.5757667
ニアレストネイバー法 (図 14(a))	7.5757667
バイリニア法 (図 14(b))	7.6581029
バイキュービック法 (図 14(c))	7.7862561
Lanczos(4)法 (図 14(d))	7.7960045

4.2 グレイスケール画像とカラー画像の比較

24bit カラー画像 (図 2) をグレイスケール化し画像補間法で拡大した 8bit グレイスケール画像 (図 17) と 42bit カラー画像 (図 3) をグレイスケール化し画像補間法で拡大した 16bit グレイスケール画像 (図 19) を, 元のカラー画像を画像補間法で拡大した画像と比較を行う. 使用する画像は本研究室で撮影した画像を使用している.

まず、24bit カラー画像（図 16）と 8bit グレイスケール画像（図 17）の比較を行う。

主観評価では、カラー画像とグレイスケール画像に違いは色以外確認できなかった。客観評価で出したエントロピ（表 2）では原画像とニアレストネイバー法に関しては、情報量の変っていないことがわかる。ニアレストネイバー法以外の画像補間法を用いた拡大画像では、カラー画像のエントロピが大きくなっており、情報量の違いが確認できる。カラー画像とグレイスケール画像の両方とも、バイリニア法、バイキュービック法、Lanczos(4)法の順に大きくなり、情報量の増えていることが確認できる。原画像の画素値のほかに多数の画素値を補間し、画像補間法ごとに参照する画素の数異なるためだと考えられる。



(a)ニアレストネイバー法

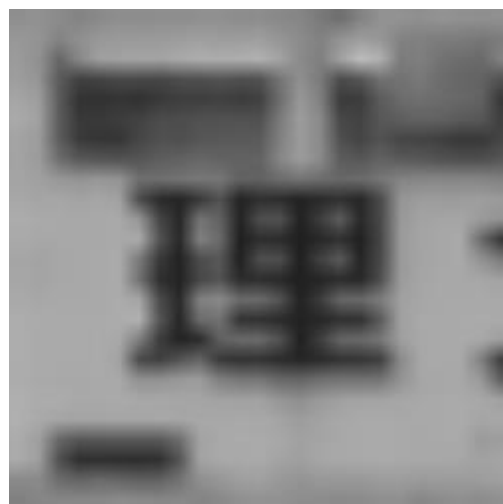


(a)ニアレストネイバー法



(b)バイリニア法

図 16((a),(b)) 24bit カラー画像

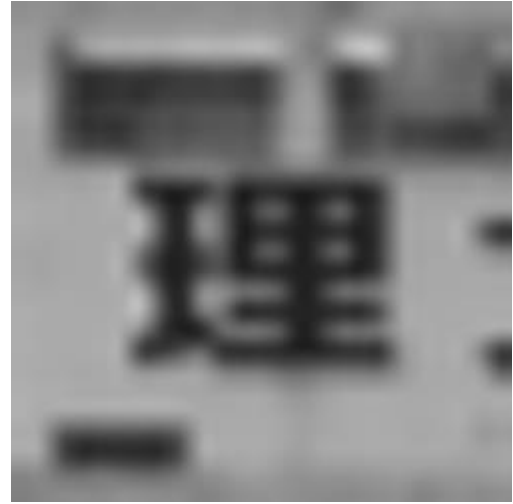


(b)バイリニア法

図 17((a),(b)) 8bit グレイスケール画像



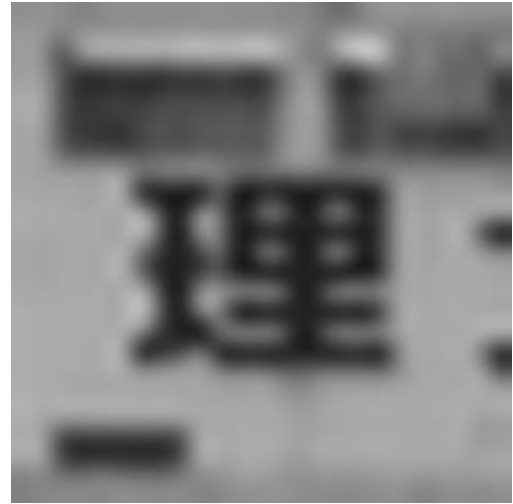
(c)バイキュービック法



(c)バイキュービック法



(d)Lanczos(4)法



(d)Lanczos(4)法

図 16((c),(d)) 24bit カラー画像

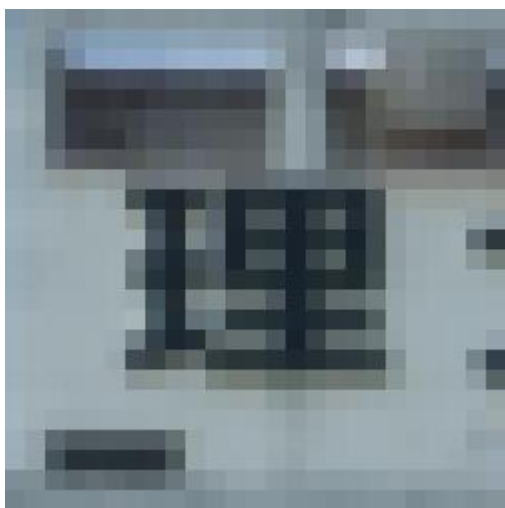
図 17((c),(d)) 8bit グレイスケール画像

表 2 24bit カラー画像と 8bit グレイスケール画像のエントロピー

	24bit カラー画像 (図 16)	8bit グレイスケール画像 (図 17)
原画像 (図 2)	6.6592705	6.6592705
ニアレストネイバー法	6.6592705	6.6592705
バイリニア法	6.8047957	6.8026486
バイキュービック法	6.8924068	6.8887425
Lanczos(4)法	6.8994024	6.8988842

次に、42bit カラー画像（図 18）と 16bit グレイスケール画像（図 19）の比較を行う。本来 16bit グレイスケール画像と比較すべき bit 数は 48bit となるが、本研究では 48bit カラー画像ではなく RGB 各 14bit 計 42bit のカラー画像を使用している。

主観評価では先ほどと同様、カラー画像とグレイスケール画像の違いは色以外確認できなかった。客観評価で出したエントロピ（表 3）では原画像とニアレストネイバー法のエントロピはグレイスケール画像の数値が大きいことがわかる。ニアレストネイバー法以外の補間法ではグレイスケール画像のエントロピが小さいことがわかる。ニアレストネイバー法はそれぞれの原画像のエントロピと同じ値になり、原画像の情報量と比べても変化していないことがわかる。



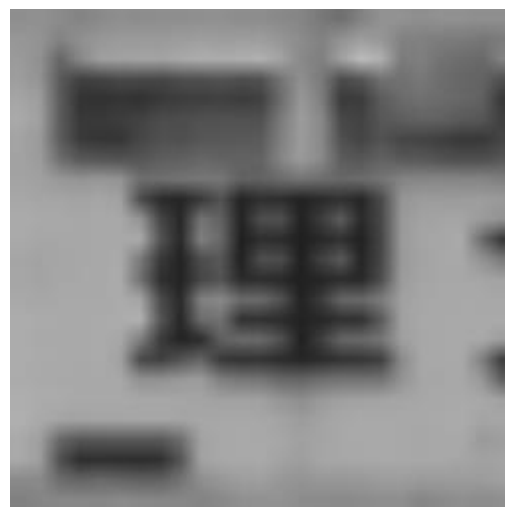
(a)ニアレストネイバー法



(a)ニアレストネイバー法



(b)バイリニア法



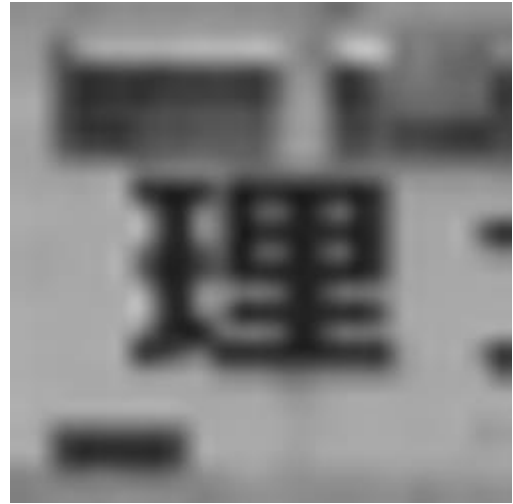
(b)バイリニア法

図 18((a),(b)) 42bit カラー画像

図 19((a),(b)) 16bit グレイスケール画像



(c)バイキュービック法



(c)バイキュービック法



(d)Lanczos(4)法



(d)Lanczos(4)法

図 18((c),(d)) 42bit カラー画像

図 19((c),(d)) 16bit グレイスケール画像

表 3 42bit カラー画像と 16bit グレイスケール画像のエントロピー

	42bit カラー画像 (図 18)	16bit グレイスケール画像 (図 19)
原画像 (図 3)	6.6458393	6.6592705
ニアレストネイバー法	6.6458393	6.6592705
バイリニア法	6.8066627	6.8020851
バイキュービック法	6.8965202	6.8889808
Lanczos(4)法	6.9048454	6.8975345

4.3 8bit グレイスケール画像と 16bit グレイスケール画像の比較

8bit グレイスケール画像 (図 17) と 16bit グレイスケール画像 (図 19) の比較を行う。主観評価では大きな違いはみられなかった。客観評価で出したエントロピー (表 4) では、原画像と 4 通りの画像補間法を用いた画像のエントロピーを比較する。原画像のエントロピーを比べると情報量は同じであり、ニアレストネイバー法を比べると同じ数値になっている。バイリニア法と Lanczos(4)法では 8bit グレイスケール画像のエントロピーが大きく、バイキュービック法のみ 8bit グレイスケール画像のエントロピーが小さいことがわかる。

表 4 8bit グレイスケール画像と 16bit グレイスケール画像のエントロピー

	8bit グレイスケール画像 (図 17)	16bit グレイスケール画像 (図 19)
原画像 (図 2), (図 3)	6.6592705	6.6592705
ニアレストネイバー法 (図(a))	6.6592705	6.6592705
バイリニア法 (図(b))	6.8026486	6.8020851
バイキュービック法 (図(c))	6.8887425	6.8889808
Lanczos(4)法 (図(d))	6.8988842	6.8975345

4.4 24bit カラー画像と 42bit カラー画像の比較と評価

24bit カラー画像 (図 16) と 42bit カラー画像 (図 18) の比較を行う。

主観評価では大きな違いはみられない。客観評価では、原画像と 4 通りの画像補間法を用いた画像のエントロピーを比較する (表 5)。24bit カラー画像と 42bit カラー画像の原画像のエントロピーを比較すると、異なる数値になっている。原画像において 24bit カラー画像の数値が大きいことから、24bit カラー画像の情報量が多いことが考えられる。ニアレストネイバー法は原画像と同じ数値になっており、情報量に変化はみられない。ニアレストネイバー法以外の画像補間法は、42bit カラー画像に関しエントロピーが大きくなり、情報量の多いことがわかる。

表 5 24bit カラー画像と 42bit カラー画像のエントロピー

	24bit カラー画像 (図 16)	42bit カラー画像 (図 18)
原画像 (図 2), (図 3)	6.6592705	6.6458393
ニアレストネイバー法 (図(a))	6.6592705	6.6458393
バイリニア法 (図(b))	6.8047957	6.8066627
バイキュービック法 (図(c))	6.8924068	6.8965202
Lanczos(4)法 (図(d))	6.8994024	6.9048454

5 Lanczos(n)法の違い

3.4節で述べたように Lanczos(n)法 (式 9) は n に任意の数を入れることによって参照する画素の数を変更することができる。本研究では $n = 4$ で拡大を行ったが、ここでは $n = 2, 3, 4, 5$ としてそれぞれ重み関数を計算した。倍率を s とすると原画像の画素の座標を q 、補間したい座標を $s\Delta p$ 、重みの総計を t とする。重みの総計は1に近いものとなる。本研究では倍率を4倍にしたので、補間したい座標は3個である。

Lanczos(4)法のグラフ (図 12) から分かるように、 $W(x)$ 軸を中心に左右対称になっていることから下記の表 (表 6, 7, 8, 9) も対称となっている。例えば、表 6 の原画像と補間画素の重みを $(4\Delta p_1, q_{-1})$ と $(4\Delta p_3, q_2)$ 、 $(4\Delta p_2, q_{-1})$ と $(4\Delta p_2, q_2)$ 、 $(4\Delta p_3, q_{-1})$ と $(4\Delta p_1, q_2)$ は同じ数値になっている。そのため0と1の間を中心として対称になっていることが確認できる。Lanczos(n)法の (式 9) の n の値が増加することによって、原画像の画素の座標と補間したい座標が同じでも重みの絶対値は大きくなっていることが確認できるが大きな差ではないため、拡大画像に大きな違いはないと考えられる。しかし n の値が増加することによって参照する画素数は多くなるため、拡大画像の情報量が多くなると考えられる。

表 6 Lanczos(2)法の重み

		q					t
		-1	0	1	2		
$4\Delta p$	1	-0.085	0.877	0.235	-0.018	1.010	
	2	-0.064	0.573	0.573	-0.064	1.019	
	3	-0.018	0.235	0.877	-0.085	1.010	

表 7 Lanczos(3)法の重み

		q						t
		-2	-1	0	1	2	3	
$4\Delta p$	1	0.030	-0.133	0.890	0.270	-0.068	0.007	0.997
	2	0.024	-0.135	0.608	0.608	-0.135	0.024	0.994
	3	0.007	-0.068	0.270	0.890	-0.133	0.030	0.997

表 8 Lanczos(4)法の重み

		q								t
		-3	-2	-1	0	1	2	3	4	
$4\Delta p$	1	-0.015	0.056	-0.155	0.895	0.283	-0.092	0.032	-0.004	1.001
	2	-0.013	0.060	-0.166	0.620	0.620	-0.166	0.060	-0.013	1.002
	3	-0.004	0.032	-0.092	0.283	0.895	-0.155	0.056	-0.015	1.001

表 9 Lanczos(5)法の重み

		q										
		-4	-3	-2	-1	0	1	2	3	4	5	t
$4\Delta p$	1	0.009	-0.030	0.070	-0.162	0.897	0.289	-0.104	0.047	-0.018	0.002	0.999
	2	0.008	-0.033	0.081	-0.182	0.626	0.626	-0.182	0.081	-0.034	0.008	0.999
	3	0.002	-0.018	0.047	-0.104	0.289	0.897	-0.162	0.070	-0.030	0.009	0.999

6 まとめ

本研究で比較した画像補間法に関し、主観評価ではニアレストネイバー法とバイリニア法よりも、バイキュービック法と Lanczos(4)法が鮮鋭な画像が得られている。数値による客観評価では本研究で使用した拡大画像すべてにおいて、ニアレストネイバー法を除き、バイリニア法、バイキュービック法、Lanczos(4)法の順にエントロピが増加していることが確認できる。ニアレストネイバー法は原画像と同じ数値になるため、新たな画素を作らない拡大方法であるといえる。したがって、イラストを拡大する際に新たな画素を作りたい場合はニアレストネイバー法が最適である。人物などを撮影し画像を拡大する際、輪郭などをはっきり表現したい場合は高画質な画像が得られる Lanczos(n)法が最適である。

Lanczos(4)法のエントロピがほかの画像補間法と比べて最大であることから、拡大画像の情報量が多くなり高画質での表現が可能であることがわかる。Lanczos(n)法で使用される n の値を増加させることにより、参照する画素が多くなるため情報量が多くなる。そのため n の値が大きければ大きいほど拡大画像は高画質となるはずであるが、わずかな差であるため、目視での確認は困難であると考えられる。

参考文献

- [1] C 言語による画像処理プログラミング入門, 長尾智晴, 株式会社 昭晃堂, 2011 年
- [2] 画像拡大法の改善, 竹崎仁美・山崎茜, 岡山理科大学総合情報科学卒業論文, 2013 年度
- [3] OpenCV2 プログラミングブック, OpenCV2 プログラミングブック制作チーム, 株式会社 マイナビ, 2011 年