

# Lagrange Lanczos (n) 法による拡大画像について

澤見研究室

1131012 江田匠 1131014 大嶋圭悟 1141067 錦織慶亮

## 1 はじめに

近年では SNS など写真や画像を見たり用いる機会が増えており、その画像を加工するソフトも多く出ている。本研究では画像を加工する中で画像の拡大について焦点をあて、既存で存在する画像の拡大方法より原画像に近く画像を拡大できる方法はないのかとの興味、関心から、今回は既存の方法より優れた補間法として新たに LagrangeLanczos 法を提案することにした。

## 2 原画像について

原画像としてラスタ画像を使用した。主な形式は png (Portable Network Graphics), jpeg (Joint PhotoGraphic Experts Group), bmp (Bitmap) に分類され、一般的なデジカメやスマートフォンでは jpeg が用いられている。ラスタ画像は別名ビットマップ画像ともいわれ、デジタルカメラやインターネットで用いる画像として用いられており、Photoshop などの編集ソフトでも用いられている。一般に 3 原色それぞれを 8 ビットで表しドットピクセルを 2 次元配列にして構成された画像のことで、一般的なディスプレイを用いて色や濃淡の変化を自然に見せる場合に適している。



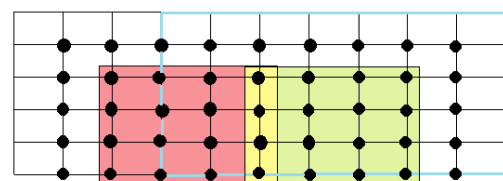
図 1 ラスタ画像の見本

## 3 補間の手法について

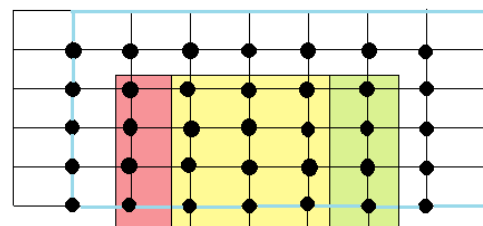
補間の手法にはブロック単位ごとの補間とポイント単位ごとの補間がある。ブロック単位ごとの補間とは、まず原画像から大きさが  $n \times n$  の画素ブロックを取り出し、このブロック内の点上ですべての画素を補間したら次のブロックに移り、これを画像全体に対して繰り返して行う。このとき 1 次多項式なら  $2 \times 2$  の画素ブロックから 4 ピクセル単位、3 次多項式なら  $4 \times 4$  の画素ブロックから 16 ピクセル単位でブロック内の補間計算をする。

ポイント単位ごとの補間の場合は、ブロックの移動や範囲の指定がブロック単位の場合と異なり、ブロック境界のピクセルを 1 ピクセル単位ずらしながら補間していく。

3 次多項式の場合のそれぞれの補間方法を示す。赤いブロックが 1 回目の補間、緑のブロックが 2 回目の補間、黄色のブロックが重複して補間された部分である (図 2)。



(a) ブロック補間



(b) ポイント補間

図 2 横方向にブロックを重複させた補間

## 4 画像補間法

画像補間法は画像の拡大,縮小,回転,変形などを行うときに用いられており,補間の際ピクセルとその周辺のピクセルの間に中間ピクセルを計算により生成する技術である.

主な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,sinc Lanczos 法などがあるがここでは sinc Lanczos 法と新たに提案する補間法の Lagrange Lanczos 法について調べていくことにする.

### 4.1 sinc Lanczos(n)法

sinc 関数とはフーリエ変換を用いた補間方法に使われており,正弦関数をその変数で除して得られる初等関数のことをいう.画像の拡大をする際に画像が劣化してしまう原因はいくつかあるが,そのひとつに処理の過程で起こる誤差振動がある.この誤差振動が起こることによって画質は劣化し,その振動が大きいほど画質の劣化も大きくなる. sinc 関数はグラフで見ると中心点から遠ざかるにつれて振動の幅が小さく緩やかになっていくが,振動がなくなることはない(図2).これがギップス現象とよばれる画像劣化の原因となっている.

sinc Lanczos (n)法は sinc 関数を重みづけをすることで補間誤差の振動を極力減らした sinc Lanczos 関数を用いてピクセルを補間計算し画像を拡大する方法である. sinc 関数と比べても誤差振動は少なくなっておりギップス現象の影響は軽減されている. Lanczos 法はピクセル数を  $2n*2n$  個単位のブロックで補間するため,ニアレストネイバー法などと比べると計算量が多くなるため処理速度はかなり遅くなる.しかし次数  $n$  を大きな数値に変更することで拡大時の画像の粗さが減少し,画像を高画質で拡大することができる. sinc Lanczos 法による拡大は主に  $n=3$  が用いられている.  $n$  の値が大き

なりすぎると誤差も大きくなっていくため,かえって画質が低下してしまう.

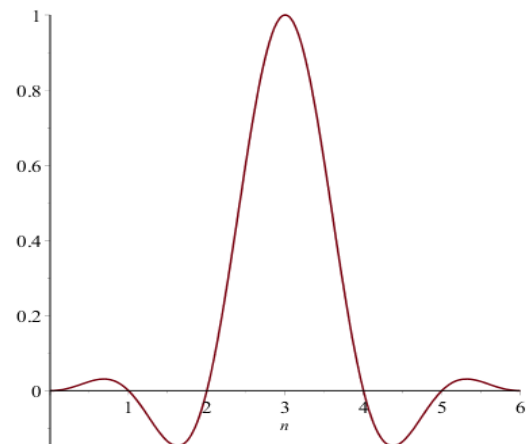


図2 sinc(n=3)の sinc 関数

### 4.2 Lagrange Lanczos(n)法

画像の補間法の重みづけに, sinc 関数ではなく Lagrange 多項式を用いてピクセル計算する補間法がある.次数  $n=3$  程度であれば偽象によって画質が大きく劣化するという事は少ない.次数  $n$  が大きい多項式を用いる場合には,ブロック単位で処理すると画素ブロックの端のほうの計算において大きな誤差振動が発生する場合があります.これをルンゲ現象という.

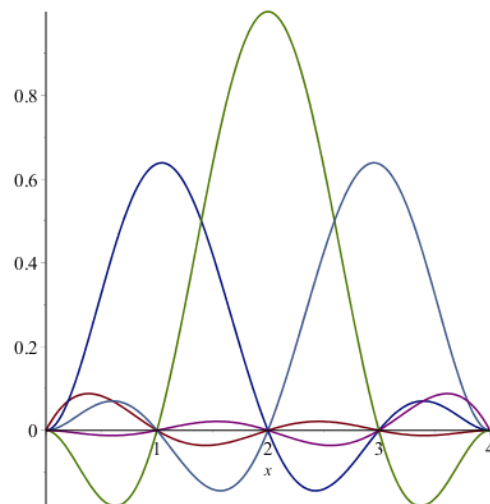


図3 Lagrange Lanczos(4)の関数

## 5.1 手法による比較

原画像をブロック単位とポイント単位の補間によって拡大し, 比較と評価を行う. 評価の方法としては, 目視による主観評価と, 評価値を求めて行う客観評価がある. 今回は主観評価と, エントロピとして計算した数値に比較による客観評価を行う.

表 1(a) ブロック単位補間の画像エントロピ

補間次数	画像エントロピ
4	6.8526738
8	6.5371206
16	5.9499044
原画像	6.7541581

表 1(b) ポイント単位補間の画像エントロピ

補間次数	画像エントロピ
4	6.6943993
8	6.4396536
16	5.8842059
原画像	6.7541581



図 4 4次ブロック

図 5 4次ポイント



図 6 8次ブロック

図 7 8次ポイント

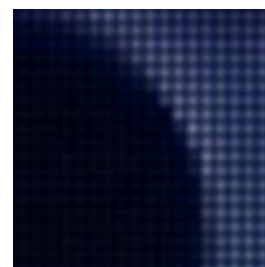


図 8 16次ブロック 図 9 16次ポイント

原画像にそれぞれ 4 次, 8 次, 16 次のブロック間またはポイント間の補間を行い, その一部を切り取った画像を示す(図 4, 5, 6, 7, 8, 9).

画像エントロピによる客観評価を行った結果, エントロピによる数値の比較では 4 次ではポイント単位の補間がより原画像に近い数値となっているが, 8 次と 16 次ではブロック単位の補間のほうが原画像により近い数値となった(表 1). 目視による主観評価では, 同じ次数での補間を比較すると, 4 次の補間ではそれほど大きな差は見受けられないが(図 4, 5), 次数が 8 次, 16 次と高くなるにつれ画質の劣化が大きくなっていくのが分かる(図 6, 7, 8, 9). 特にブロック単位では画像の劣化が格子状に大きく表れており, ルンゲの現象が起きていることが分かる(図 4, 6, 8). 特に 16 次のブロック単位の補間では色の偏りが激しくかなりの劣化となっている(図 8). 主観評価ではややポイント単位の補間が優れていると分かった. これらの結果からエントロピの数値はブロック単位の補間が多少上回っているものの, 実際の拡大画像の画質としてはポイント単位の補間の方が計算量が同じで画質が優れていると結論づけることができる.

## 5.2 補間法ごとの比較

ラスト画像(図 1)を前述した sinc Lanczos 法と Lagrange Lanczos 法で拡大しそれぞれを比較する. 輪郭の曲線部分は劣化が分かりやすくなるため, 赤枠で囲まれた目の部分を拡大する(図 10-a).



図 10-a 拡大前の画像



図 10-b sinc Lanczos(3)法での拡大



図 10-c Lagrange Lanczos(4)法での拡大

表 2 補間法ごとの画像エントロピ

画像補間法	画素情報量
sinc Lanczos(3)法	6.6261339
Lagrange Lanczos(4)法	6.7546139
原画像	6.7541581

目視による主観評価では大きな差は見受けられないが、画像エントロピによる客観評価では Lagrange Lanczos 法の画素情報量が原画像

にかなり近い数値を出しており、優れていると結論付けることができる。

## 6. まとめ

ブロック単位の補間とポイント単位の補間では、ポイント単位での補間の方が優秀であるという結論が出た。今回の補間に用いたプログラムでは、簡単のため画像の端の部分の計算処理をしていないので、端を切り取って画像の比較を行っている。実用化するためには、端の部分も正確に処理する必要がある。また既存の補間との比較によって Lagrange Lanczos 法による補間が、一番原画像に近い画質を維持しつつ拡大できることが分かった。しかし客観的評価として画素情報量が 1 番近いということではしか評価していないため、ほかの評価方法を利用した比較をすることも今後の課題とする。

## 参考文献

- [1] Visual Studio2017 に OpenCV3.2.0 と opencv\_contrib を導入する方法  
<https://qiita.com/tomochiii/items/fa26404ebc5fcd4481b9>
- [2] Visual Studio 2017 + OpenCV 3.2.0 + x64 の初期設定  
<http://phst.hateblo.jp/entry/2016/11/30/081749>
- [3] 画像補間法による拡大, 田中・仲, 岡山理科大学卒業論文 2015 年度
- [4] 補間法による拡大画像の比較, 神崎・大霜・坂本, 岡山理科大学卒業論文 2016 年度
- [5] 画像の解像度変換について, 三宅勇輔, 岡山理科大学修士論文 2016 年度