

ビジュアル情報処理採取課題レポート

学籍番号 : 1710021

氏名 : 有馬海人

テーマ設定理由

プログラムに関しては、「講義の課題・宿題の拡張でも良い」とあったので、提出した課題につけられたコメントで最も指摘の多かった「hw07-1」の課題を選定した。指摘点の多かった課題を選定した理由は、改善すべき点が多いために指摘が多くなったと考え、改善点を修正することで「宿題の拡張」になると考えたからである。

プログラムの仕様・工夫

コメントには

バイラテラルフィルタでは赤緑青それぞれのチャンネルに対して専用のフィルタをかけていきます。ですので配列fをf[3][][]と拡張してあげましょう。h(・)の中の1つ目のexp()は例題7.13のgaussian関数で求めているsumのようにすればできます。これを2つ目のexp()とかけ合わせればh(・)になります。exp()の中は各チャンネルごとに輝度値の差分を求め、それを $2\sigma^2$ で割った値になります。後は、filtering関数のようにf(・)h(・)の総和を求め、h(・)の総和で割ればバイラテラルフィルタになります。

とあった。

ただ、このコメント通りに修正を重ねると [ビジュアル情報処理 第9回 宿題プログラム例](#) との差異が無くなりオリジナリティが失われると考えたため、自分が宿題として提出したものにコメントにつけられた修正点を適用していくという方針でプログラムを作成していくことにした。

ここで、達成目標として以下の2点を設定した

1. 次の2式の明確な実装

$$g(i, j) = \frac{\sum \sum f(i+m, j+n)h(i, j, m, n)}{\sum \sum h(i, j, m, n)}$$

$$h(\cdot) = \exp\left(-\frac{m^2 + n^2}{2\sigma_1^2}\right) \exp\left(-\frac{(f(i, j) - f(i+m, j+n))^2}{2\sigma_2^2}\right)$$

2. 元プログラムでは輝度値を用いていなかったため、輝度値を用いたプログラムとする

勉強した点

1. Color型の引数に使用可能な関数の調査

- 今まではcolor型を引数とする関数は `red()`, `green()`, `blue()` をすでに知っていたが、輝度を求める手段を知らなかったため、color型の引数を取り輝度を返す関数である、`brightness()` を見つけた

2. バイラテラルフィルタそのものについての調査

- 授業内資料だけでは自分には理解が及ばなかったため、WEB上でバイラテラルフィルタについての調査を行った。調べた結果、バイラテラルフィルタはガウシアンフィルタを輝度差に基づいて重みづけをしているということが分かった。このことから、以前に宿題として提出したコードでは赤・緑・青の差をとっていたため全く間違っていたということが分かった。

3. 二乗を行う関数について

- 計算時に二乗を行うのだが、以前のプログラムでは一文字の変数にのみ二乗を行っていたのだが、今回修正するにあたって、二乗を行う個所の記述が長くなってしまったため、二乗を行う `sq()` 関数を使用した。この関数を用いることで簡潔に書くことができた。

技術的ポイント

プログラムを作成するうえでの技術的ポイントを述べる。

1. $h(i, j, m, n)$ を実装する際に大きく二つに分けた。

- 関数 $h(\cdot)$ は $\exp(\cdot) \cdot \exp(\cdot)$ というように記述することができるので、プログラム内では二つに分けた。こうすることで、プログラムが横に長くなりすぎることを抑制し、可読性を上げることができた。

2. 関数 $g()$ について

- 赤・緑・青それぞれに対して専用のフィルタをかけるとのことであったので、 $g()$ のプログラム内での出現量は高くなる。しかし、関数 $g()$ を丸々とそのままプログラム内に記述すると、関数 $h()$ の時と同様に横に長くなってしまいかねないという点に同じような記述が増えてしまうと考えた。この問題を解決するために関数を分子と分母に分け、必要になったらそれらを割り算するという形式をとった。こうすることで、被っている部分は一つにまとめることができた。

考察

- バイラテラルフィルタをかけた第一段階は「画像中で色が大きく変化するエッジ部分ではその色の違いが保存されるように平滑化が施され」ていることが非常にわかりやすい結果となった。このことから、私が修正したバイラテラルフィルタのプログラムは課題通りの動作をしていると考える。
- 設定した二つの達成目標は二つとも達成できたと思う。一つ目に関しては関数 $g()$, $h()$ の両方を分割などを行っているとはいえわかる形で実装することができたと思うからである。また、二つ目に関しては実装時に `brightness()` を用いているため、確実に達成することができたと思う。

参考文献など

- processing英語リファレンス (https://processing.org/reference/color_datatype.html)
- イメージングソリューション / バイラテラルフィルタ (<https://imaging-solution.net/imaging/bilateralfilter/>)

- 画像の本質を引き出す / バイラテラルフィルタによるノイズ除去とイラスト風画像の生成 (https://www.jstage.jst.go.jp/article/itej/62/8/62_8_1268/_pdf)
- Processingクイックリファレンス (<http://www.musashinodenpa.com/p5/index.php?pos=364>)