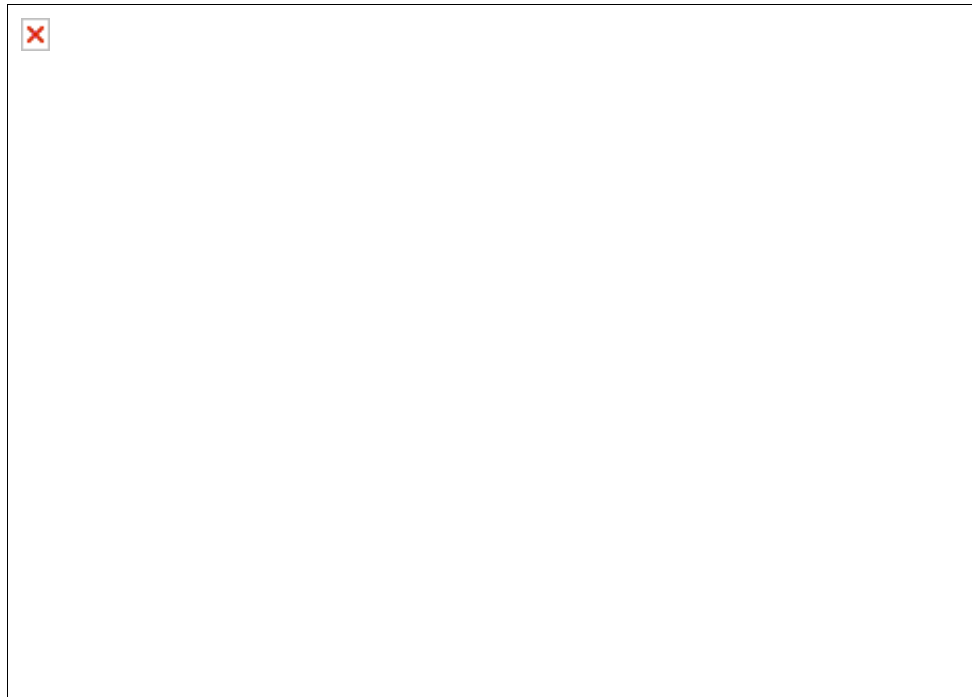


～画像処理ソフトの製作～

小島 大典



今回の発表で、私は「画像処理ソフト作り」と、「画像処理」の研究成果を報告いたします。

まず、「画像処理」の定義から。

1: 画像処理とは・・・

「画像処理」とは何でしょう？まずは、この言葉の意味をはっきりさせておきましょう。参考に、フリーの百科事典サイト「ウィキペディア」から引用・・・短くまとめたものを載せておきます。

↓

画像処理(がぞうしより)とは、**画像**から何らかの**情報**を取り出すために行われる処理全般を指す。

画像とは、コンピュータ上のデータの中で、特に、何らかの形、色を表現、保持しているものをさす。また、単に点の集まりをさすこともある。

たとえば、写真などの画像は、形と色を保持しているし、スケッチなどの画像も、同じように形と色を保持している。

画像処理の流れの一例として、画像入力ー画像変換ー分類 というものを挙げる。まず対象とする画像を入力し、この画像に対して変換処理を行う。変換の方法としてはさまざまなものが存在するが、基本的なものでは濃淡画像を白黒2値にする「2値化」、濃度変化から物体の境界を見出す「**エッジ検出**」などがよく用いられる。これらいくつもの変換処理を重ねて行うことで必要な情報の抽出を行い、最後に得られた情報の分類を行う。

…うーん。。

ま、要は「画像」という写真などのデータを、あっちゃこっちゃ弄って、目的のデータにする作業というわけです。

たとえば、写真を一部分切り取ったり(トリミング)、明るさを調節したり(ヒストグラムの調整)、縦横の大きさを変えたり(リサイズ)といった作業が相当します。

そして、この画像処理を潤滑に、便利に進めていけるようなツールを「画像処理」するソフトというわけで、「画像処理ソフト」と呼んでいるわけです。これから私が発表する「画像処理ソフト作り」も、そういったソフトを作る過程をまとめたものでございます。

さて…画像処理の説明をしましたので、これを実現するためのアプローチをこれから追って説明します。

2: 画像…??

画像…と一言で言いますが、なんのこっちゃですね？

先ほどは「画像」の説明をしましたが…

「画像とは何だろう？」

…難しく考えても答えは出てきません。

皆さんがパソコンを使う際に見続けている「モニター」ですが、すごく近くで見ると、同じ大きさの小さい無数の四角が集まっていることに気づくでしょう。そして、パソコンを遠くから見るとひとつの画面…ひとつの大きな「絵」として見る事が出来ます。

つまり、色のついた点の集まったもの全ては「画像」と呼べるわけです。

パソコンでたとえば写真の編集をするかと思えます。そんな時、同じようにモニターの中の写真を近くで見ると、色の違う点の集まりだと気づくでしょう。

これからも分かるように、画像とは「点の集まり」だと言えます。こう考えると、画像処理とは、画像…つまり有限な大きさ(最小単位の点ひとつ)の集まりに何らかの処理をして、目的の画像にすること…つまりつまり、点一つ一つに何らかの処理を加えて、点一つ一つを目的の点に作り変える作業だといえます。

これを踏まえて、コンピュータ上で画像処理をするために何をすればいいのかを次にお話します。

3: 画像をデータにする…

画像の説明をしました。では、コンピュータ上で画像をどう扱うのかをお話します。

…とその前に。。

コンピュータでは、皆さんもご存知のように、全てを「2進数」でまかっています。つまり、0と1ですね？この2進数を使って、足し算をしたり…写真を編集したり…ゲームをしたり…と、全ての作業はこの0と1だけで行っているのです。すごいね。

ですが、この0と1だけではほとんど意味を持ちません。「0と1がたくさん集まっている」から、意味を持つのです。

たとえば…

皆さんは足し算をします。その際は、単に「足せばいい」わけです。難しくありません。

コンピュータでも、足し算するときには、「足せばいい」わけですが…「2進数」という部分で、ちょっと事情が違ってきます。

たとえば、皆さんが足し算をするときは、「0から9」までの数字を使うわけですが、この数字ひとつでは、足し算は出来ませんよね？2個以上…つまり、「桁」という概念で持って、足し算などや掛け算などを行っているわけです。

コンピュータでも、「桁」が全てを決めます。桁の概念があるから、「数字」を扱え…「数学」を扱え…そしてそこからあらゆる作業を行えるようになるのです。桁は大切なのです。さて、コンピュータではこの「桁」のことを「ビット」と呼んで区別しています。桁数をビットの数で表現し、たとえば5桁なら「5ビット」などと呼んでいます。

このビットは、皆さんが良く使う数の概念と全く同じです。たとえば、0から9の数字を使うと、1桁で10種類の数、2桁で100種類の数、3桁で1000種類の数を区別できますが、区別できる数の個数は「10のn乗」とあらわせます。2進数では、0…1…ときたらすぐ次の桁になりますので、扱える組み合わせの数は

n ビット… 2 の n 乗

とあらわせますね？

さて、話がそれてしまいました。

コンピュータで画像を扱う際には、「明るさ」と「色」を扱わないといけません。だって…モノクロじゃ寂しいでしょ？そこで、色をコンピュータで扱うために、色を「数字」で扱ってみましょう。

皆さんご存知のように、「明るさ」とは、見かけが明るい暗いかをさす言葉です。たとえば、夜は「暗い」し、昼は「明るい」ですね？

まずはこれを、コンピュータで扱ってみましょう。

と、その前に…明るさって、大きさがあるものなのではないでしょうか？

実は、コンピュータでは、「無限」という概念は扱えません。なぜなら、無限が「数字ではない」からです。数字でないものを皆さんも数字では表せず、「 ∞ 」という言葉で一緒にしてしまっています。無限には「大きさ」がないので、区別が出来ません。コンピュータでは、全てを2進数で扱うの

で、「区別できないもの」は扱えないのです。

…ではどうしようか。。

そこで、無限を「近似する」ことを考えます。

たとえば、明るさを数字の大きさであらわすとします。このとき、細かく細かく見ていくことは出来ませんが、先ほどの通り、無限小の明るさはコンピュータでは表せません。そこで、「人間の限界」を利用します。

人間は、ある程度の明るさがないと、明るさを明るさとして感じる事が出来ません。これは、人間の構造なので仕方ない。これは、「最小限の明るさ」といえます。

この最小限の明るさ以下は、人間が区別できないので、考えてもしようがありません。扱っても意味が無いなら、この明るさを1としたときに、その何倍の明るさなのか…それで明るさを扱えるようになります。つまり、明るさを数字で扱えるようになったわけです。

具体的には、人間は大体「256段階」位の明るさの変化を感じることが出来ます。これは、2進数で言うところの2の8乗…つまり「8ビット」ですね？

この8ビットの数字で明るさを扱えるようになりました。無論、何ビットで扱っても、明るさは明るさですが…。

次に、「色」を考えましょう。

皆さんが今見ている光景は「カラー」ですよ？白黒しか見えない人なんてこの中にはいませんよね？

では、コンピュータでもこの「色」を扱えなければいけません…ではここである原理のご登場。

皆さんも効いたことがあると思いますが、全ての色は「光の三原色」という、「赤・青・緑」(の明るさ)の組み合わせであらわせることが分かっています。

つまり、この3色の組み合わせで全ての色が表現できるわけです。

では、ここでようやく「色」があつかえるようになりました。先ほどの明るさの概念を、赤青緑3つに適用すれば、いいわけです。つまり、8ビットのデータが3つになりました。

ここにきて、ようやく画像を表現するための「最小単位の点」を扱えるようになりました。

さて…先ほど言ったように、画像とは色のついた点の集まりでした。点の集まりの中での点ひとつひとつは、「座標」という概念で特定できます。つまり、左から何番目…下から何番目の点なのか？ということです。

先ほどの点は、3色の色の組み合わせであらわせました。そして、座標ごとの色を…つまり、一個一個の点の色を決めると、遠目で「絵」のように見る事が出来るようになります。

・・・ちょっと急いじゃいましたが、画像をコンピュータ上で扱えるようになりました。

4: 画像処理をする・・・

画像を扱えるようになりましたので、いよいよ画像処理をして見ましょう。

画像処理は、「点一つ一つ」を弄ることだといいました。では、実際の画像処理をどう行えばよいのでしょうか？

たとえば、画像処理のひとつに「明るさ調整」というものがあります。これは、画像の見かけの明るさを調整する処理で、写真などを明るくしたり暗くしたりする処理です。

皆さんの生活に当てはめると、たとえば「電気をつけたり消したり」することが相当します。夜の部屋で電気がついてなければ何も見えないですが・・・スイッチをつけて部屋を明るくすることが出来ます。

画像も、時には暗すぎて何も見えないから明るくしたい時だってあります。そんなときに使われるのがこの処理なのです。

黒板をご覧ください。

この処理はきわめて簡単です。明るさを上げるのは、明るくする・・・暗くするのは暗くする・・・。つまり、明るさを変える・・・数字を上げ下げする処理・・・一言で言えば、足し算引き算だけで済むのです。

これは、点一つ一つに目的の数字を足しひきしてあげればよいのです。

こんな感じに、意外と画像処理は単純な方法で済むことが多いのです。これを応用すると、たとえば、逆行写真の逆行部分だけを検出して、必要な明るさに調整したりも出来ますが、原理は足し引きだけなのです。

また、他にも星の数ほど画像処理の方法が研究されています。その一例は黒板を見てください。

5: ソフトを作る・・・

ここからは、少し難しい話になりますが、追って説明いたします。

画像処理ソフトは、画像処理を行うソフトだといいました。では具体的にどういうソフトなのでしょう？

以下の図をご覧ください。

画像処理のAtoZをフローチャートで説明しました。これを最後にソフトにすればいいわけです。そして、上のフローチャートの重要部分を数字ごとにまとめましたので、以下の図をご覧ください。

これを、プログラミングで作成します。「VC++」というソフトで作りましたが、詳しい内容はただのプログラムの羅列になりますので、今回は割愛させていただきました。その際、ソフト内部での画像データの流れを図に示しましたので、以下の図をご覧ください。

これで、ソフト作りが完了しました。

6:最後に・・・

今回、ページが足りずに、はしょって説明してしまった部分もありましたが、ソフト作りの研究成果を何とかまとめられました。最後まで長々と読んでくださり、ありがとうございました。

引用:ウィキペディア「画像処理」

<http://ja.wikipedia.org/wiki/%E7%94%BB%E5%83%8F%E5%87%A6%E7%90%86>

私のホームページ「森の汽笛隊」

<http://morik.net/>

私のメールアドレス

m@morik.net

今回のソフトの置いてあるページ

<http://labo.morik.net/soft/dl/>