

画像解析による 「ご飯の量自動推定アルゴリズム」の開発

小川 祐介

Yusuke OGAWA

名古屋文理大学 情報文化学部 情報文化学科 はせがわ研究室
HASEGAWA Laboratory, Department of Information Culture, Nagoya Bunri University

平成17年1月28日 提出

要旨

写真画像のデータに処理を施すことによって、目で見ただけでは読み取れない情報を引き出せる可能性がある。そこで今回は、「ご飯」のデジタル写真画像から「ご飯の量」を自動で推定するアルゴリズムを考えた。このアルゴリズムが実現できれば、栄養管理が必要な患者やダイエットを気にするユーザが、外食の際などでも食事の写真を撮るだけで、ご飯の量が分かるシステムなどに応用でき、大変有用であると考えられる。

写真画像から情報を抽出するにあたって、対象物の抽出（画像のどの部分がご飯なのか）情報の抽出（ご飯の量をどのように推定するか）の問題があり、これらのアルゴリズムの開発に取りくんだ。まず、JAVA 言語を使用して画像から各画素の色データをRGBの値で抽出し、対象物である「ご飯」の領域を、色データを手がかりにして、画像中の指定した領域内から割り出す。そして「ご飯」と認識された部分の面積、横幅、縦幅などから、ご飯の量を推定した。本研究では、推定アルゴリズムとして3通りの方法を考案し、それぞれの方法について、ご飯の量の測定値と計算結果を散布図にして最小二乗法による近似曲線の当てはめをおこなって推定に必要な定数を定め、実際にご飯の量を推定してその精度を評価した。評価の結果、考案したアルゴリズムのうちの1つは、用意した8枚の画像において、ご飯の量を実用可能な程度まで正確に推定することができた。画像から対象物の量を算出するアルゴリズムは、今後、ご飯だけでなく、多種の対象物の自動推定に拡張し、適切な機能を加えることにより、利用性の高いシステムに応用できる可能性を持っている。

1. はじめに

今日、ネットワーク社会において様々な情報・データがやり取りされている。このネットワークにより伝達されるものは、文字や数

字、音声、そして画像と多種多様に及ぶ。一般に、これらのデータを直接見ることや聞くことで情報を取得することができる。しかし画像データにおいては、目で見ただけでは読み

取ることのできない情報も画像に処理を施すことで引き出せると考えられる。

今回は、食事療法が必要な在宅患者の栄養指導を画像データを使っておこなう場合や、患者が摂取カロリーを簡単に自己管理できるようにするため、ご飯のデジタル写真画像から必要なデータ(ご飯の量)を自動推定するアルゴリズムを考えた。開発したアルゴリズムの内容と推定精度の評価の結果を報告し、また、画像の自動解析アルゴリズムの可能性について考察する。

2. 実験

本研究では、「ご飯」のデジタル写真に画像処理を行うことによって「ご飯の量」を推定して自動算出するプログラムを作成し、はじめから量が分かっている「ご飯」の写真に適用して、推定精度を評価する実験を行った。

2.1. 実験環境

一般に、写真画像から目的の情報を抽出するには、画像のどの部分が対象物なのか、および、見えない情報をどのように抽出するか、が問題になる。

今回は、画像データを処理できるクラスライブラリが豊富にあり、動作環境がプラットフォーム(使用するOSなどの環境)に依存せずブラウザがあれば誰にでもネットワーク上で動かすことのできるアプリケーションとするためJAVA言語を使用し^{1)・3)}、器に盛られたご飯の画像をもとに、ご飯の領域を抽出し、ご飯の量を自動推定するプログラムを作成した。

2.2. 解析に用いた画像

ご飯を、同じ種類の器に50g、100g、150g、200gずつ盛ったもの(図1参照)を、器の種類とバックの色を変えてサンプルAとB(A1~4とB1~4)の2セット(合計8杯分)用意した。これらを、器がだいた

い画面の中心に来るようにそれぞれデジタルカメラで撮影して、図2のような画像を8枚用意した。この画像に処理を行った。

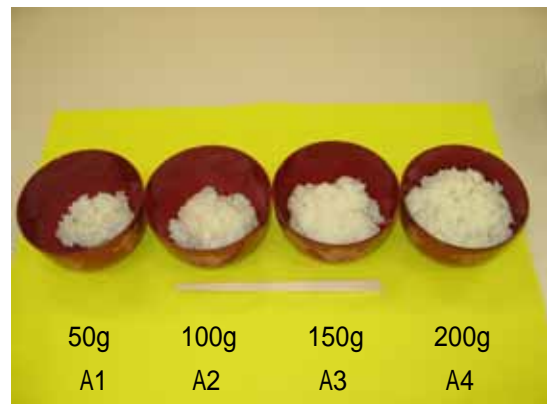


図1 実験に用いたご飯(サンプルA1~4)



図2 実験に用いた画像の1つ(B3:150g)

2.3. 対象物の領域抽出

ご飯(対象物)の量を測定するには、まず画像のどの部分が「ご飯」なのかを判断しなければならない。器に盛られたご飯の形は一定でないため、輪郭や形でご飯の部分を割り出すのは困難である。そこで、色による対象物識別方法を試みた。

用意した写真画像(図2のような8枚)に写っているご飯の色のR,G,B値を調べると、R 97, G 97, B 97,であった。よって画像中のご飯の領域抽出において「ご飯の色は上記の範囲内である」ことを一つの条件とした。ただし、この条件だけでは箸の色などとの区別が困難であったため、さらに条件を追加した。色条件をクリアするピクセルのうち、画面の中央付近に写った「ご飯」

```

int setPix(){
    int rr,gg,bb;
    int cnt = 0 ;
    for(int y=0;y<height;y++){
        for(int x=0;x<width;x++){
            rr = ((pix[y*width+x] & 0x00ff0000)>>16) & 0x000000ff ;
            gg = ((pix[y*width+x] & 0x0000ff00)>>8 ) & 0x000000ff ;
            bb = ((pix[y*width+x] & 0x000000ff)>>0 ) & 0x000000ff ;
            if( rr>97 && gg>97 && bb>97 && // 「ご飯」領域 色 円内
                (x-width/2)*(x-width/2)+(y-height/2)*(y-height/2)<10000){
                cnt ++ ; // 領域計測 領域の面積 最大・最小座標
                if(x<x_min) x_min = x ; if(y<y_min) y_min = y ;
                if(x>x_max) x_max = x ; if(y>y_max) y_max = y ;
                new_pix[y*width+x] = 0xff0000ff ; // 領域を青色に
            }else{
                new_pix[y*width+x] = pix[y*width+x] ; // 領域外は元画像
            }
        }
    }
    return( cnt ) ;
}

```

図3 「ご飯の量自動推定プログラム」のソースコード (JAVA アプレット) の一部

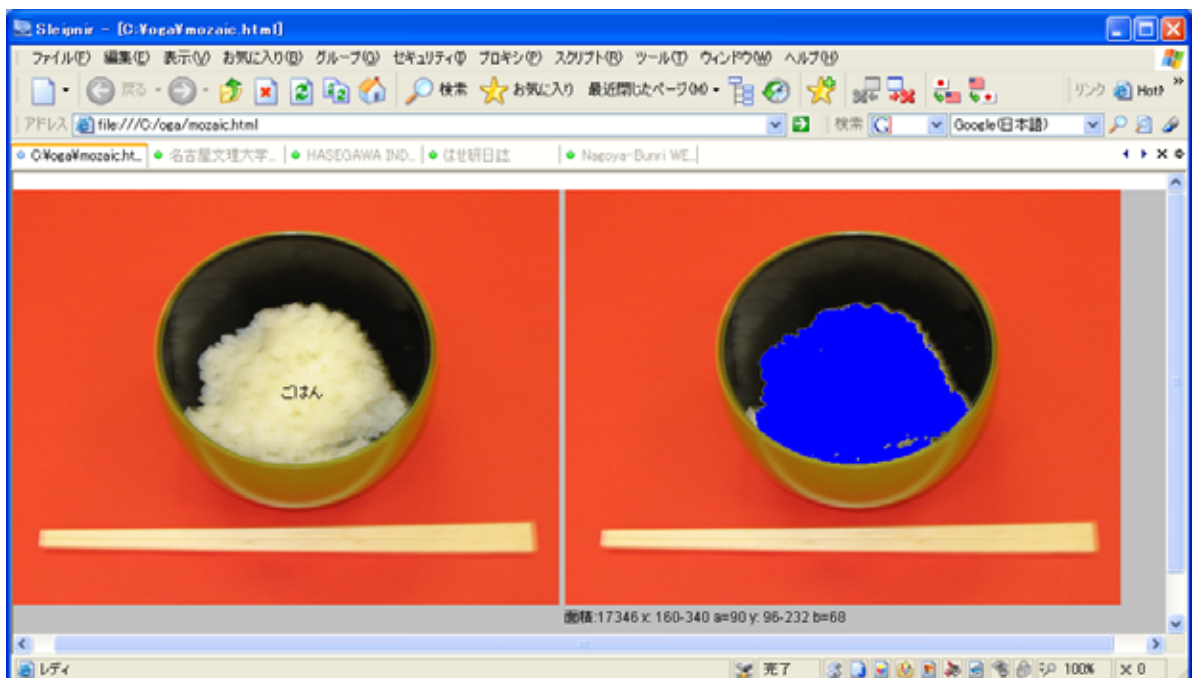


図4 「ご飯の量自動推定プログラム」の実行例 (元画像 (左) と出力画像 (右))

だけを、木の割り箸などと区別するため及びノイズの影響を除くため、画像処理の対象になる領域を円 (中心座標と半径) で指定することを条件に加えた。つまり指定領域内の対象色をご飯とみなす方法で、対象物の領域抽出を行った (図3 参照)。

対象物の領域抽出において、上記の方法の有効性を確認するため、「ご飯」と認識した箇

所を青色に変換するプログラム (図3) とし、このプログラムで、用意した8枚の画像すべてを試した。図4のように、元の画像 (左側) のうち「ご飯」と認識された領域が、処理画像 (右側) に青くなって出力される。

この出力結果 (図4) から、指定領域内の対象色をご飯とみなす方法で、ほぼ狙い通りの領域の抽出ができていることを確認した。

2.4. 画像計測

次に画像からご飯の量を推定するために必要だと思われるデータを計測する。今回考えた3つの推定方法(後述)の中で必要なデータは、1) ご飯の面積、2) ご飯の横方向の半径、3) ご飯の縦方向の半径、4) 割り箸の長さ、の4つである。1)は、抽出した領域のピクセルの数をカウントし、2)3)は、抽出領域の x 座標値または y 座標値の最大値からそれぞれの最小値を引き2で割ることで求めた。4)は、箸の長さ方向のピクセル数をカウントして求めた(図3・図4参照)。

求められた値の単位はすべてピクセルだが、ご飯の量を推定するために、単位を cm に変換する。単位の変更には、画像から計った割り箸の長さ(ピクセル)と実際の割り箸の長さ(cm)の比率を用いた。割り箸の長さは一般的なものを11本調べた結果、 20.2 ± 0.4 cm であったため、単位の変更には 20.2 cm を用いた。割り箸は同じでも8枚の写真画像から計った割り箸の長さ(ピクセル)は同じではないので、比率(cm /ピクセル)は画像ごとに異なる。計測した値(ピクセル)の単位を cm に変更した値を、それぞれ、次のように表す。

$$\left. \begin{array}{l} \text{ご飯の面積 (cm}^2\text{)} = S \\ \text{ご飯の横方向の半径 (cm)} = a \\ \text{ご飯の縦方向の半径 (cm)} = b \end{array} \right\} \text{(式1)}$$

このように計測した S, a, b を用いて、ご飯の量を推定した。

2.5. ご飯の量の推定

前述の計測値 S, a, b から、ご飯の量 W (g) を推定する方法を、以下の3通り考えた。

(1) 推定方法

ご飯の量 W (g) は、その体積 V (cm^3) に比例すると考えられる。 V は、器に盛られたご飯の形で決まる。底面の半径と高さとともに r cm の逆円錐 $V = 1/3 \pi r^3$ (cm^3) と、半径 r cm の半球 $V = 2/3 \pi r^3$ (cm^3) の中間

の形と仮定すれば、 V は、 πr^3 に比例すると考えられる。以上のように考え、ご飯領域の横方向の半径 a (cm) を使って、ご飯の量を次式で算出する。

$$W = \alpha \pi a^3 \quad \text{(式2)}$$

(2) 推定方法

画像上で抽出したご飯の部分が楕円形に近いことから、楕円の面積 πab (cm^2) をもとに、次数を3乗にして立体の体積を推定できると考え、ご飯の量を次式で算出する。

$$W = \alpha \pi (ab)^{\frac{3}{2}} \quad \text{(式3)}$$

(3) 推定方法

画像は斜め上からの写真であるため、ご飯の部分は楕円形のような形に写っているが、もし器の真上から撮影したならばご飯は円に近い形に写ると予想できる。よって真上から見たときの面積は、面積の計測値 S に a/b をかけた値に近いと考えられる。これにもとづいて、ご飯の量を以下のように算出する。

$$W = \alpha \pi^{\frac{1}{2}} \left(\frac{a}{b} S\right)^{\frac{3}{2}} \quad \text{(式4)}$$

式2・3・4とも、 S, a, b は、式1に示した計測値、 α は定数である。

図5・6・7に、計測した8つの画像について、式2・3・4の定数 α を含まない計算結果(縦軸)を各サンプルの量(横軸)との散布図に表示し、その近似曲線(原点を通る1次直線)をサンプルAとB毎に示す。これにより、 α の値を求めた。推定方法(式2)では $\alpha = 1/2.0416$ (サンプルA)、 $1/2.1687$ (サンプルB)(図5参照)、推定方法(式3)では $\alpha = 1/1.3783$ (A)、 $1/1.5614$ (B)(図6参照)、推定方法(式4)では $\alpha = 1/1.8308$ (A)、 $1/1.8169$ (B)(図7参照)となった。これらの α 値を使って求めた3つの方法(式2・3・4)による推定の結果を表1・表2に示す。また、それぞれの方法の推定精度を誤差の少なさで評価した。

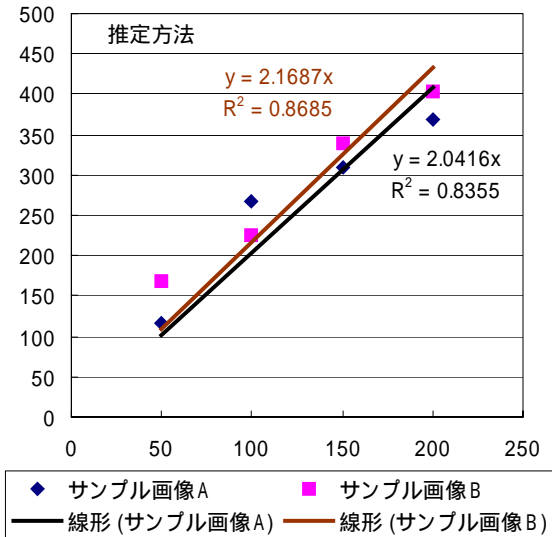


図5 推定方法 散布図

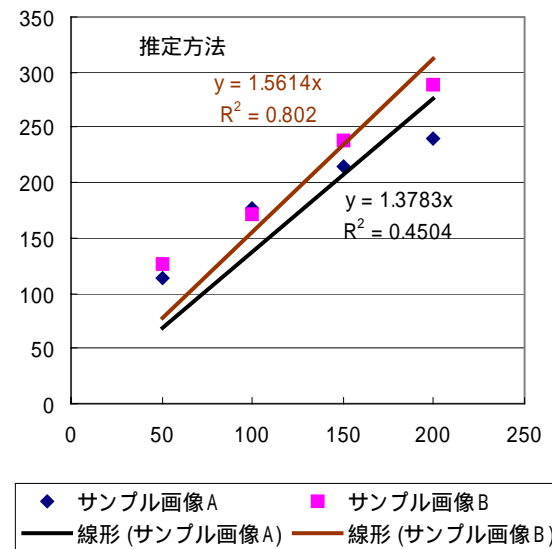


図6 推定方法 散布図

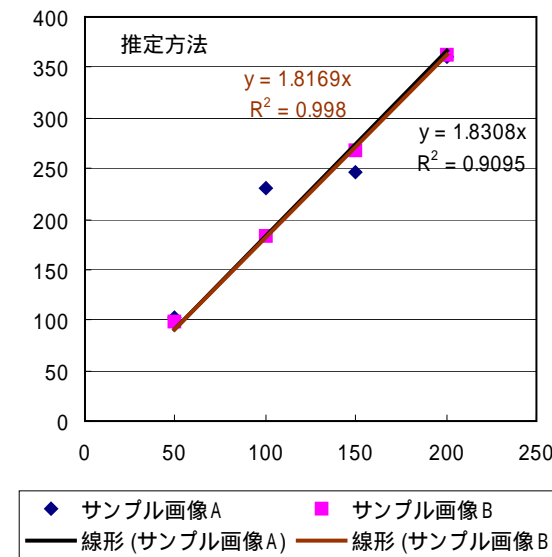


図7 推定方法 散布図

表1 推定結果 (サンプルA)

サンプル画像	A1	A2	A3	A4
ご飯の量 (g)	50	100	150	200
推定方法	57.3	131.3	151.8	181.2
推定方法	82.7	127.7	156.0	173.4
推定方法	55.7	125.6	134.9	197.2

表2 推定結果 (サンプルB)

サンプル画像	B1	B2	B3	B4
ご飯の量 (g)	50	100	150	200
推定方法	77.0	103.3	156.8	186.7
推定方法	80.7	109.5	153.1	185.7
推定方法	53.9	101.3	147.5	199.8

3. 結果

各推定方法 (式2・3・4) の定数を除いた計算結果は、図5・6・7に示したとおりである。近似曲線の一致度は R^2 値 (相関係数 R の2乗値) で判断できる。この R^2 値は1に近づくほど近似の精度が高いことを表す。今回の推定方法の中では $R^2 = 0.998$ となった推定方法のサンプルBが最も直線近似の精度が高かったことを示している。

また、近似直線の傾きからを定めて算出した「ご飯の量」の推定値 (表1・表2) を見ると、推定方法では、特にA1・A2・B1で実際の量より多く推定されている。推定方法ではA2・A3で誤差が大きいものの、他のサンプルでは、実際の量と良く一致しているといえる。50gあたりの平均的誤差 (g) を式5のように、それぞれ求めた。

$$err_{50} = \sqrt{\frac{1}{N} \sum_{n=1}^N \left\{ \frac{50}{w_n} (W_n - w_n) \right\}^2} \quad (\text{式5})$$

ただし、 w は各サンプルのご飯の量 (真値)、 W は推定値、 n はサンプル番号1~4、各サンプルの数 N は4である。

推定方法では、 $err_{50} = 8.9$ (サンプルA)、 13.7 (B)、同様に、方法では、 18.1 (A)、 15.6 (B)であったのに対し、方法では、 7.5 (A)、 2.0 (B)と誤差が小さかった。

4. 考察

栄養管理やカロリー計算のためには、ご飯の量 50 g あたり数 g の誤差は許容範囲であると考えれば、今回開発したアルゴリズムのうち、推定方法は、今回使用したサンプル画像において、ご飯の量をほぼ実用可能な程度まで正確に測定できたと評価できる。

また、今回開発したような画像解析アルゴリズムは、ご飯以外の多種の対象物の自動推定や他の機能を加えることにより、利用性の高いシステムに応用できる可能性がある。たとえば、プログラムに栄養素の種類やカロリーの計算システムを追加すると、食事制限を必要とする人が外食をする際、デジタル写真を撮るだけでその場で食事の分量と栄養バランスが分かり、食事管理が手軽にできる。

画像から対象物の量を算出する技術は、応用しだいで非常に利便性の高いものになる可能性がある。今回、その実現性の一部を実際に示すことができた。

5. 今後の課題

今回のアルゴリズムは、お碗型の容器に盛られた「ご飯」を対象に開発・評価したが、他の形の容器（たとえば図 8 (a) のような平皿）の場合についても、式 4 の定数を変えただけで高い精度の推定ができるかどうかは、今後確かめる必要がある。また、今回のアルゴリズムは、ピクセルの色データによって、ご飯（対象物）を抽出するものである。そのため、抽出の対象物の部分と対象物ではない

部分のピクセルの色が同じ場合（たとえば、図 8 (b) のように器も白い場合）や対象物の色が一定でなかったり見えない場合（たとえば、図 8 (c) 三色丼）両者を自動で識別することができない。また、対象物の色（RGB 値）や対象物の抽出領域は写真画像の写し方によって差がある。今回作成したプログラム(図 3)では、色領域・対象物抽出領域を制限する作業は、プログラムを変更することが必要である。今後、操作性の向上が望まれる。

さらに、今後はご飯だけでなく他の料理などの分量も、画像から自動で確実に判断できるようにするためのアルゴリズムの開発が課題である。

謝辞

本研究にあたって、サンプル画像を提供いただいた、名古屋文理大学健康生活学部健康栄養学科の奥村万寿美先生、照井眞紀子先生に謝意を表します。

参考文献

- 1) 杉山三樹雄：「JAVA ではじめる画像処理プログラミング」、株式会社ディー・アート(2002)
- 2) 大塚真・西村勇亮：「JAVA 先生」、毎日コミュニケーションズ(2003)
- 3) 加藤智也：紅葉処理アルゴリズムの開発、はせがわ研究室卒業研究報告 2002, 名古屋文理大学情報文化学部はせがわ研究室、p.2-1 ~ 2-4、(2003)



(a) 平皿の場合



(b) 白い茶碗の場合



(c) ご飯が見えない(三色丼)

図 8 自動解析が困難と思われる画像の例