

画像解析による葉面積計測

Tree leaf area measurement by Image analysis application

共通事業本部 環境部

平田 康明

植物の葉は光合成生産の基本となるが、その大きさは、活性や発生時期、位置等の生育条件により大きく変化する。本稿では、剪定の影響による樹木の葉の大きさの変化について、葉面積を指標として把握するにあたり、スキャンした画像を画像解析ソフトimageJ(Fiji)により葉面積の計測を行った事例を紹介する。

1 はじめに

葉面積は「一枚の葉の面積」であり、植物の光合成を行う生産の基本を示す指標となるが、同じ個体の中でもその生育条件により大きさが異なる。

剪定が樹木に与える影響を検証するための調査において、前年度の剪定強度が異なる樹木について葉の大きさの違いを検証する必要がある。

本稿では、一枚あたりの葉面積を個別かつ多数計測することを目的とし、スキャナ機器により取得した葉画像について汎用PCとフリーの画像解析ソフトを用いて葉面積を計測した方法について述べる。

なお、植物の群落構造の研究や生産量の指標としては、葉面積指数[LAI=Leaf Area Index:単位面積あたりの葉面積(片面) 総計]が用いられる。LAIの計測では刈り取りによる個別の葉面積計測という直接的かつ多大な労力を伴う手法を経て、近年では簡易計測が可能なLAI計測機器が開発されている。これらは非破壊的な手法であり、透過光量やセンサーを用いた機器や、全天魚眼レンズを用いて解析を行うプラントキャノピーアナライザ等があるが、本稿での手法の目的と異なるため詳細は言及しない。

2 葉面積の計測方法の選定

2.1 葉面積の計測手法

葉面積一枚ごとの計測法としては、従来、方眼に写し取り面積を計数する方法、葉を紙に写し取って切り抜き重量計測により面積換算する重量法、葉の長さや幅から近似式により求める方法等が用いられてきたが、面積計測自体に多大な労力が必要であった。

近年ではデジタル機器の発展により、以下の新しい方法が開発されている。

① 葉面積計

葉面積を比較的簡便に計測することを目的とし

た計測製品。現地もしくは室内で葉の表面を走査し、面積を計測するもので、ハンディタイプ、据え置き型、海外製・日本製の製品が購入可能である。(写真2.1参照)

② スキャン画像と画像解析ソフトによる計測

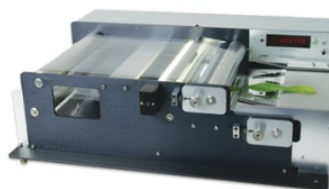
スキャナ機器により取得した葉画像を用い画像解析ソフトにより面積計測を行う。



CI-202 ポータブル葉面積計 (CID社製)



AM350 葉面積計 (英弘精機株式会社製)



LI-3100C 卓上葉面積計 (LI-COR製)

写真 2.1 葉面積計の製品例
(写真はいずれも製品HPより転載)

2.2 計測手法の選定

本稿では葉面積計は専用機器としての扱いやすさやデータロガーによるデータ出力可能等、機能性は高いものの高価であり、コスト的に一時使用には見合わないと判断した。

スキャン画像と画像解析ソフトによる計測では、スキャンイメージの取り込みには汎用のスキャナ機器を用いることができ、解析には汎用PCとフリーの画像解析ソフトを使用することで導入コストは抑えられる。

葉面積計測が可能で、現時点でダウンロード可能なフリーソフトとして、「LIA for Win32 (LIA32) ¹⁾」をまず検討した。

LIA32では、ピクセル解析による葉面積計測は容易なもの、解析は1画像ごとの手作業となることから本稿の目的のような大量の画像処理には向かないと判断した。(写真2.2参照)

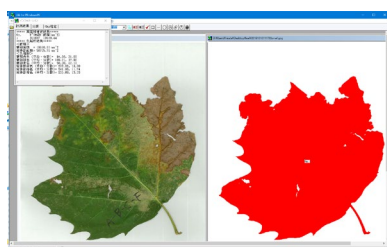


写真 2.2 LIA32による計測例

その他のフリーソフトウェアとして画像解析ソフトimageJ (Fiji) ^{2) 3) 4) 5)}を検討し、以下の利点から本ソフトを用いることとした。

- ・解析機能として葉面積計測に対応し、解析パラメータ設定の自由度が高い。
- ・各種プログラミングマクロによる一括処理が可能である。
- ・研究分野で広く使われ、オープンソースの為にプラグインも豊富で、WEB情報も多く得られる。
- ・パブリックドメインの為、将来的な有償化や開発中止のリスクが低い。

*imagej: アメリカ国立衛生研究所(NIH)でWayne Rasbandにより1997年に開発された画像処理ソフトウェア。Java仮想マシン上で動作するマルチプラットフォーム。

imagej (Fiji)はImageJに様々な機能(プラグイン)を同梱して配布したパッケージ。

3 計測の手順

採取から葉面積計測までの手順を図 3.1 に示す。

3.1 サンプルの下処理

サンプリングした葉は、取り込みまでの一時保管の間、

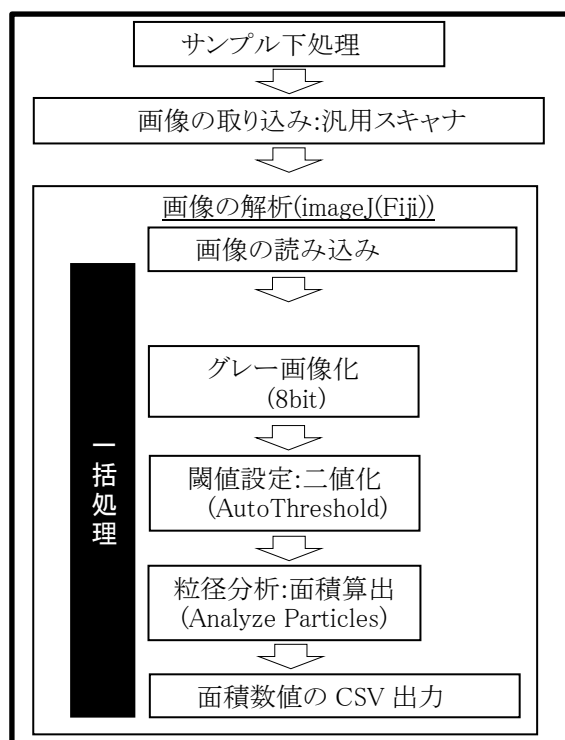


図 3.1 imageJ (Fiji)による葉面積計測の手順

葉の萎縮等によるスキャン不良を防止する為、新聞紙等を用いて「押し葉」状に処理した。比較的短期間で取り込み処理が可能な場合は完全に乾燥させる必要はないが、活性が高い夏の葉は水分量が多く、長期保管するためには、十分な乾燥が必要となる。葉の質によっては乾燥により脆くなる場合もあるので取り扱いにも留意が必要である。

3.2 画像の取り込み

サンプリングした葉はフラットベッドタイプのスキャナ(A3サイズ)で画像取り込みを行った。

画像取り込み形式は、高解像度のTIFF形式やデータ量が軽減されるJPG形式等が使用できるが、本ソフトはTIFFのマルチページの解析には対応しておらず、事前に別のアプリケーションで単独ファイルに分離することが必要である。

解析ではA3に収まりきらない大型の葉もあり、これらは適宜断裁・分割して取り込みを行い、面積計測後に合計処理を行った。

画像解析のオプションでは、取り込み画像の白背景・黒背景を選択できるが、一部の枯れ・変色・陰などの発生により白背景画像での処理はエラーとなりやすい為、黒色の板を使用して黒背景で処理を行った。なお、スキャナの蓋を閉めずにスキャンを行うことで黒背景となるので、スキャン時に十分な葉の平面性が保てる場合は開閉作業を省力化できる。

3.3 葉面積の計測

画像解析では、画像の「読み込み」⇒「グレー画像化(8bit化)」⇒「閾値(二値化)」⇒「粒径分析」の処理により面積値が得られる。(画像3.1参照)

例示したように、色味や陰影に関わらず、ノイズ・虫食い等を除外した良好な解析結果が得られた。なお、葉柄部は除外していないが、画像3.1の例で葉身部の1.6%程度の面積であり、誤差内と見なせるであろう。

解析で用いた「グレー画像化(8bit化)」、「閾値(二値化)」、「粒径分析」の処理内容とオプション・パラメータは以下の通りであり、サンプルの条件によって最適な結果が異なるため、適宜調整することが望ましい。

グレー画像化(8bit化) …元画像からモノクロ8bit(256階調)画像として二値化の基準値を単一化する。

閾値(二値化)…グレー画像から白黒2値の画像化により認識領域と非認識領域に分割する。

Manu:[Image]>[adjust]>[threshold]

- ・閾値:プリセットの選択が可能。試した中で最も適した「Triangle」を選択した。
- ・dark:黒背景設定。(図3.2参照)

粒径分析…認識領域外郭を生成して領域面積が出力される。

Manu:[Analyze]>[Analyze Particles]

- ・size:認識領域ピクセル面積の最小-最大サイズ。最小値以下の領域外のノイズ除去ができる、葉のサイズにより調整。
- ・circularity:円形あるいは非円形に近い領域を選択(設定無し)。
- ・show:認識領域を表示
- ・exclude:エッジ連結部の除外。映り込んだ不要な周縁部を除外できる。
- ・include:領域内の穴部を認識する。size設定値に準じて領域内のノイズや虫食い等の欠損部を除外する。

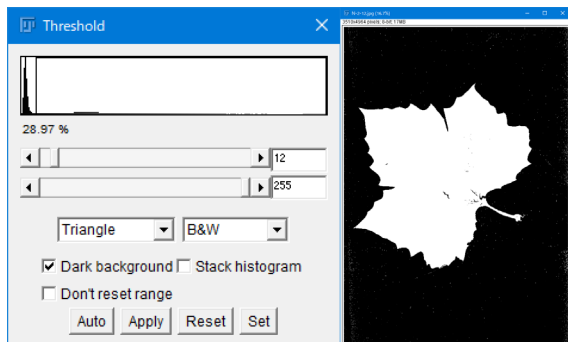


図 3.2 閾値(二値化)のオプション設定画面

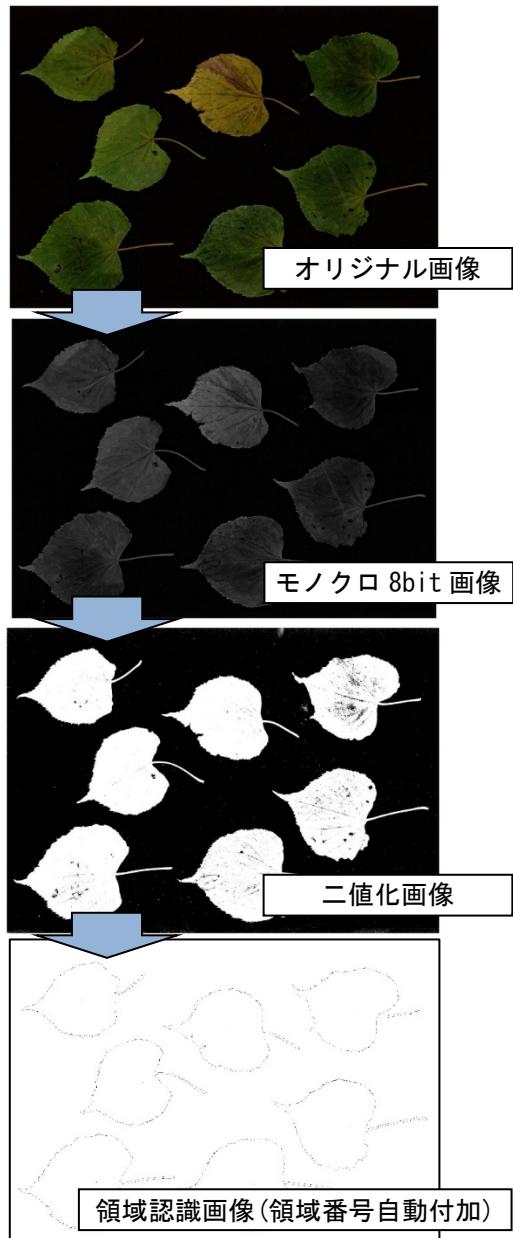
次頁に計測用に作成したマクロを示す。

解析対象画像を格納したフォルダを指定することで、当該フォルダ内の画像全ての面積計測までを一括で処理し、領域認識部ごとの面積値(「元画像名」+.csv)(表3.1参照)とチェック用の領域認識画像(「元画像名」+ap.jpg)(画像3.1参照)を出力する。

表 3.1 画像解析結果の出力例

	Area	Mean	Min	Max
1	10.08	39.408	12	179
2	6.666	49.535	12	192
3	7.547	57.141	12	178
4	7.86	67.179	12	184
5	9.928	39.572	12	201
6	8.312	34.448	12	161
7	7.89	28.955	12	157

*認識領域ごとに面積(赤枠部)を計測して出力(単位はinch²)



画像 3.1 画像解析の処理過程

```

#python macro 2byte文字のpath下では動作しない。赤字部は任意に設定
from ij import IJ
from ij.io import DirectoryChooser
import os
#####
#Analysis()
def Analysis(path):
    imp = IJ.openImage(path)
    savefilepath = os.path.splitext(path)[0]
#グレー画像化
    IJ.run(imp, "8-bit", "")
#2値化 パラメータ・閾値設定・背景設定
    IJ.setAutoThreshold(imp, "Triangle dark")
#面積計測 パラメータ・穴あき/ノイズ除去サイズ・認識領域表示・エッジ連結部の除外
    IJ.run(imp, "Analyze Particles...", "size=5000-Infinity pixel show=Outlines display exclude")
#解析画像の保存
    IJ.saveAs("jpg", savefilepath + "_ap.jpg")
#close images
    IJ.run("Close", "")
#面積データの書き出し(個別)
    IJ.saveAs("Results", savefilepath + ".csv")
    IJ.run("Clear Results", "")
#####
# ダイアログを開きフォルダを選択 フォルダパスを取得
srcDir = DirectoryChooser("Choose Folder").getDirectory()
IJ.log("directory: " + srcDir)
#当該フォルダのファイル名(タプル)取得 拡張子「.tif」のファイル検索 >Analysis() を実行
for root, directories, filenames in os.walk(srcDir):
    for filename in filenames:
        if filename.endswith(".tif"):
            path = os.path.join(root, filename)
            IJ.log(path)
            Analysis(path)
#面積データの書き出し(最終一括) [面積データの書き出し(個別)]の2行を削除 以下2行(#)を削除
#IJ.saveAs("Results", srcDir + "result.csv")
#IJ.run("Clear Results", "")
IJ.log("Finish")

```

解析処理速度は対象の画像サイズとファイルアクセス速度に大きく依存するが、通常の環境であれば数百枚程度なら数分で処理が可能である。

4 まとめ及び今後の展望

本稿では、大量のスキャン画像について一括処理を行った例を紹介した。類似の解析やデータ処理の一助あるいは取り組むきっかけとなれば幸いである。

近年、小型カメラや通信技術の急速な発達・低価格化により、様々な場面で大量の画像データ取得が行われ、解析の高度化・迅速化が必要となっている。

画像解析技術についても、動画処理やAI等の認識技術と組み合わせることで日々進化しており、より高度な活用が今後、期待される。

[参考文献]

- 1) 山本一清:植物画像解析用ソフトウェア「LAIfor Win32 (LIA32)」,名古屋大学大学院生命農学研究科 生物圏資源学専攻 資源生産生態学講座、2005。
<https://www.agr.nagoya-u.ac.jp/~shinkan/LIA32/>
- 2) Rasband, W.S., ImageJ, U. S. National Institutes of Health, Bethesda, Maryland, USA,
<https://imagej.nih.gov/ij/>, 1997-2018.
- 3) Schneider, C.A., Rasband, W.S., Eliceiri, K.W. "NIH Image to ImageJ: 25 years of image analysis". Nature Methods 9, 671-675, 2012. (This article is available online.)
- 4) Abramoff, M.D., Magalhaes, P.J., Ram, S.J. "Image Processing with ImageJ". Biophotonics International, volume 11, issue 7, pp. 36-42, 2004. (This article is available as a PDF.)
- 5) Schindelin, J.; Arganda-Carreras, I. & Frise, E. et al. (2012), "Fiji: an open-source platform for biological-image analysis", Nature methods 9(7): 676-682