

緑視率自動測定のためのテンプレートマッチングを用いた建物領域の認識システム

○丁 亜奎*1 福田 知弘*2
矢吹 信喜*3 道川 隆士*4

キーワード：緑視率、景観、自動測定ツール、画像認識、テンプレートマッチング

1. はじめに

1.1. 緑視率と現状の測定方法

都市環境改善のため緑化が推進されている。緑の定量化は、緑化推進の過程で、客観的評価のために重要である。緑視率は、視界の中に占める自然の緑の割合であり、平面的にとらえる「緑被率」に対して、空間的な実感に近い指標として考えられた概念¹⁾である。本研究は、エンドユーザによる緑化推進を目指して、エンドユーザが直感的に理解しやすい緑視率を対象とする。

緑視率測定の方法は、いくつか提案されているが²⁾、一般に実用化されている手法は、いくつかの視点における測定データで代表させることである。すなわち、代表的な視点で写真撮影をした後、作業者が Adobe Photoshop のような画像処理ソフトを用いてマニュアル操作で目視により測定対象となる自然の緑部分のマスキングを行い、緑視率測定を行う（以下、「実用化測定手法」とする）³⁾。

この実用化測定手法は、マスキング時間を要すること、作業により発生する作業時間のバラつきが課題である。また、実用化測定手法は処理に時間がかかるため静止視点での測定に限られる。そのため、歩行中の緑視率の変化を測定するといった動視点への対応も困難である。そこで、著者らは緑視率自動測定システムを開発中である。

1.2. 既往の成果と課題

筆者らによる既往の成果として、効率的、即時的な緑視率の自動測定システムの開発を目指し、画像処理技術を応用した測定システムについて報告してきた⁴⁾。得られた成果は以下の通りである。緑視率を測定する画像を対象として、まずガウシアンフィルタを用いた平滑化を行い、次に、色相 (H)、彩度 (S) の値をフィルタとして緑視率を測定する。開発したシステムを用いて、複数の景観画像を使用して検証した結果、自然の緑が取りうる色相、彩度の値として、 $H=(40, 180)$ 、 $S=(0.2, 1.0)$ の値を得た。

一方、本手法の課題は、緑視率の測定対象となる樹木の幹や枝などが抽出されないことである。また、緑視率の測定対象とならない人工物の一部（窓に写り込んだ自然の緑など）が過抽出されてしまうことである。

特に、窓に映り込む自然の緑成分は、入力画像に占める面積が大きいと、緑視率の数値に与える影響が大きい。窓に映り込んだ緑色成分を緑視率の計算対象から除外す

るため、画像に含まれる建物領域を検出することを検討する必要がある。

1.3. 研究の目的

そこで本研究は、緑視率測定の精度向上を目指し、現在までに開発した自動測定システムに、画像に写り込む建物領域を検出して緑視率の測定対象からの除外を可能とする機能を開発することを目的とした。

既往研究⁵⁾⁶⁾は、携帯電話の通信エリアを事前に把握するために鳥瞰画像または街並み画像から窓領域を検出、推定しようとするものであり、アプローチは参考になる。しかし、鳥瞰または街並み画像と平面地図を用いたシステムであり、本研究が対象とするアイレベルの画像のみで窓領域を検出するシステムではない。

2. アルゴリズムの検討：テンプレートマッチング

本研究で開発するシステムは、写真画像を対象として建物領域を自動的に除外するものである。緑視率で対象となる建物領域は窓や壁であり、これらを自動的に認識することを考える。仮に窓を自動認識することができれば、窓領域内部に緑色成分を含む窓自体を計算対象から除外できる。また、壁を自動認識することができれば緑視率の計算対象から予め除外できるため、特に緑色成分を含む壁に対して有効な手段となる。

窓や壁の形状の特徴として、一般に矩形が多い。そのため本研究は、画像に含まれる矩形を認識することを考える。そのため、テンプレートマッチングを応用した。テンプレートマッチングとは、まず、テンプレート画像と呼ばれる探索対象となるオブジェクト等の画像を用意しておき、このテンプレート画像を用いて、探索領域となる入力画像とのマッチングを行う手法である。テンプレートマッチングを行うことで、用意したテンプレート画像と同じパターンが画像中に存在するかどうか、存在するとしたらどの位置にあるのかを知ることができる。

以降のシステム実装と検証のために使用したコンピュータ及びソフトウェアについて述べる。OSはWindows8.1、CPUはIntel® Core™ i7 Processor (2.93 GHz)、RAMは8.00GBであり、処理には問題ないスペックであった。また、テンプレートマッチングおよびエッジ検出のためにOpenCV (ver.2.3)を用いた。

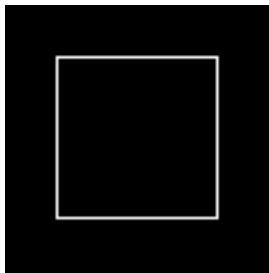


図 1. テンプレート画像

表 1. 入力画像の事前処理の検討結果 (黄色: 検出されるべき窓領域を追記)

	原画像	サンプル A (平滑化処理のみ)	サンプル B (サンプル A に Canny エッジ検出)	サンプル C (サンプル A に Sobel エッジ検出)	サンプル D (サンプル A に Laplacian エッジ検出)
入力 画像					
出力 画像					

2.1. テンプレート画像

テンプレート画像は、多様な景観画像を入力画像に対応するため、単純な矩形とした。そのため、画像サイズは 100×100 pixel、正方形のサイズは 71×71 pixel、線の太さは 2 pixel とした。

2.2. 入力画像の事前処理の検討

入力する景観画像は多様なノイズが含まれることが想定される。テンプレートマッチングにより窓以外の要素が検出されてしまうこと (誤検出)、抽出すべき窓要素が検出されないこと (未検出) をできるだけ避けたい。そのため、テンプレートマッチングに対するノイズを軽減するため、入力画像に何らかの事前処理を行うことを検討した。

まず、既報⁴⁾でも述べたように、原画像にガウシアンフィルタを用いた平滑化処理を施した (サンプル A)。次に、テンプレート画像 (図 1) は矩形 (白) と背景 (黒) との境界の輝度が大きく変化している。このような画像はエッジ検出により得ることができるため、サンプル A をエッジ検出した画像を用いることを検討した。エッジ検出手法は様々あるが、既往研究⁷⁾では Canny オペレータが画像分類に関係なく性能が高いことを指摘している (サンプル B)。Canny オペレータを実装するために、OpenCV (ver.2.3) の cvCanny 関数を用いた。cvCanny 関数で検討すべき閾値は 2 つある。大きい方の閾値は強いエッジの初期検出に利用される。また、小さい方の閾値はエッジ同士の接続に利用される。パラメータを検討した結果、2 つの閾値は 30 と 150 とした。

その他、代表的なエッジ検出手法として、Sobel オペレータ (サンプル C)、Laplacian オペレータ (サンプル D) を検討した。Sobel オペレータでは、水平方向、垂直方向の両方をエッジ検出した。

2.3. テンプレートマッチングの実装

テンプレートマッチング処理を実装する。まず、cv::Size 関数を用いて、テンプレート画像のサイズを変化させ、大きいサイズからマッチングさせる。パラメータスタディの結果、原サイズ比 0.1~3 とした。また、各画像サイズでの

マッチング個数は最大 50 個とした。これは小さな画像サイズの場合に建築ファサード要素ではないノイズを検出してしまうことを避けるためである。次に、cv::minMaxLoc 関数を用いて類似度 (0~1) の最小値を求める。パラメータスタディの結果、0.13 とした。そして、cv::matchTemplate 関数を用いてテンプレートマッチングを実行する。

2.4. テンプレートマッチング実験

2.1~2.3 で実装したテンプレートマッチングシステムを用いて、入力画像の検証実験を実施した。入力画像は大阪大学银杏会館の入口付近を正面から撮影したものとした。この画像の特徴は、自然の緑が窓面に写り込んでいる上に、窓の一部が手前の緑に隠れているため不整形の窓形状が含まれることである。

表 1 に各入力画像サンプルを用いて、テンプレート画像によりテンプレートマッチングした結果を示す。各画像で検出されるべき窓の領域を黄色で示した。各画像を比較すると、サンプル B が窓領域の抽出される結果が最も良いことがわかる。このため、入力画像の前処理として、ガウシアンフィルタを用いて平滑化した後、Canny オペレータでエッジ検出した画像を入力画像とする。

3. 景観画像を用いた検証実験




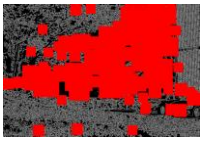
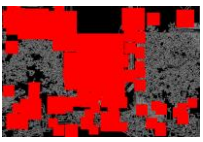
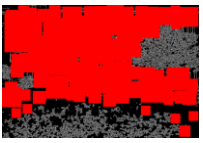
3.1. 検証用画像の作成

2 章で開発したテンプレートマッチングシステムの性能を検証するため、自然の緑や建物が含まれる景観画像を撮影して実験を行った。撮影には、一般に普及しているコンパクトデジタルカメラ (NIKON D5200) を使用して、地面からの高さ 1.5m の位置にカメラを水平に三脚で固定し、撮影した。画像の撮影日は 2014 年 6 月 11 日 (天候:曇)、撮影場所は大阪大学吹田キャンパス、撮影枚数は 37 枚 (内、3 枚を使用) であった。テンプレートマッチングの結果を表 2 に示す。

検証の方法は以下の通りである。

1. 正解画像の作成。Photoshop を利用して、原画像から、正解となる建物の窓や壁の領域を緑色に塗りつぶした。

表 2. テンプレートマッチング結果

	IMG1	IMG2	IMG3
原 画 像			
結 果			

それ以外の領域は黒く塗りつぶした。また、緑色の領域の画素数を算出した。

2. テンプレートマッチング画像の作成. テンプレートマッチングにより検出された部分を赤く塗りつぶした。それ以外の領域は黒く塗りつぶした。また、赤色の領域の画素数を算出した。
3. 二枚の画像の合成. 作成した正解画像とテンプレートマッチング画像を合成した。正解画像の緑色とテンプレートマッチング画像の赤色が重なる領域(正解領域)は黄色とした。また、黄色の部分の画素数を算出した。

3.2. 正解率と不正解率の算出

3.1 節において生成した画像に対して、正解率と不正解率を算出した。表 3 に正解率と不正解率の定義を示す。

表 3. 正解率と不正解率の定義

	定義
正解領域	検出されるべき正解領域 (表 4 画像 A の緑色領域)
正解検出領域	正しく自動検出された建築領域の画素数 (表 4 画像 C の黄色領域)
不正解検出領域	未検出領域 (表 4 画像 C の緑色領域) と誤検出領域 (表 4 画像 C の赤色領域) の画素数の和
正解率 (%)	(正解検出領域/正解領域)×100
不正解率 (%)	(不正解検出領域/正解領域)×100

算出結果を表 4 に示す。以下に本実験の結果と現システムの課題を考察する。

- 三枚の画像の正解率はそれぞれ、92.9%、76.0%、91.9%であった。IMG1 と IMG3 は高い正解率が得られたが、IMG2 は 70%にとどまった。IMG2 の正解率が低い理由は、IMG2 は窓の領域は概ね抽出されていたが、壁の領域に抽出されない部分が多かったためである。
- 三枚の画像の不正解率は 33.9%、57.5%、31.6%である。未検出は、誤検出は、であった。未検出領域では窓や壁の細かな部分を検出しきれいでなかった。また、誤検出領域では自動車、芝生、舗装などの一部が検出されてしまっていた。

4. まとめ

本研究は、緑視率の測定対象ではない窓などの建物要素を入力画像から除外するために、テンプレートマッチングを応用して、建物領域を自動的に検出するアルゴリズムを実装した。建物領域をより精度よくマッチングするため、入力画像は原画像にガウシアンフィルタを用いて平滑化した後、Canny オペレータでエッジ検出した画像を前処理で作成した。検証実験の結果、90%以上の正解率が得られた画像がある一方、70%台のものもあった。

未検出の領域、誤検出の領域に対する改善に向けたさらなる検討が必要である。

[参考文献]

- 1) 土肥博至監修, 環境デザイン研究会編著: 2007, 環境デザイン用語辞典, 井上書院.
- 2) 藤井健史, 山田悟史, 廣瀬徳郎, 及川清昭: 2013, CG モデルによる全方位緑視率の計量手法とその応用可能性, 日本建築学会技術報告集, 第 19 巻, 第 43 号, 1067-1072.
- 3) 大阪府: 2013, 緑視率調査ガイドライン, <http://www.pref.osaka.lg.jp/attach/17426/00000000/guideline.pdf> (参照 2015-10-07).
- 4) Yakui Ding, Tomohiro Fukuda, Nobuyoshi Yabuki, and Takashi Michikawa: 2015, A MEASUREMENT TOOL FOR VISIBLE GREENERY RATIO DERIVED FROM GAUSSIAN BLUR, HUE AND SATURATION FILTERING, Proceedings of the Second International Conference on Civil and Building Engineering Informatics (ICCBIEI2015), CD-ROM.
- 5) 上野智史, 明堂絵美, 酒澤茂之: 2012, ビル壁面における窓配置の規則性に注目した Web 鳥瞰画像からの窓検出, 映像情報メディア学会誌, 66(7), J236~J239.
- 6) 上野智史, 川田亮一: 2010, Web 街並画像を利用したビル壁面の窓面積比の推定手法の検討, 電子情報通信学会技術研究報告, 29-34.
- 7) 藤本伸明, 大崎美穂, 杉山岳弘, 阿部圭一: 2004, 人間によるエッジ描写結果を用いたエッジ検出手法の評価, 情報処理学会論文誌, 45(9), 2197-2206.







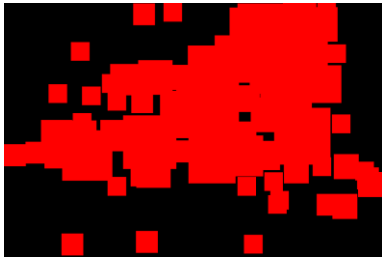
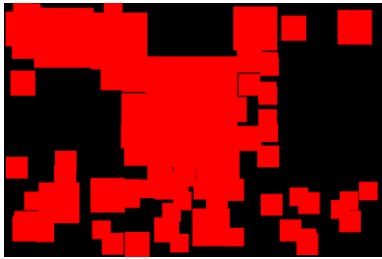

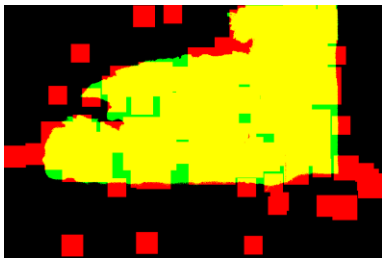
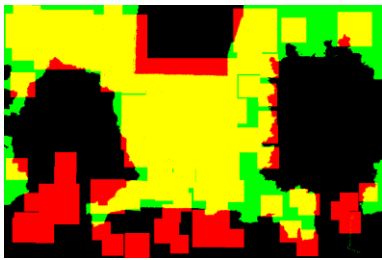
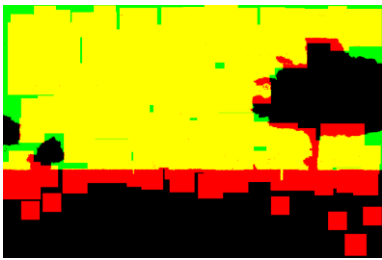
*1 大阪大学 大学院工学研究科 環境・エネルギー工学専攻 博士前期課程

*2 大阪大学 大学院工学研究科 環境・エネルギー工学専攻 准教授 博士(工学)

*3 大阪大学 大学院工学研究科 環境・エネルギー工学専攻 教授 Ph.D.

*4 大阪大学 環境イノベーションデザインセンター 特任助教 博士(工学)

表 4. 景観画像サンプルによる検証

	IMG1	IMG2	IMG3
原画像			
画像 A: 正解領域 (Photoshop で 作成) 画像 A			
正解領域の画素数	372742 pixels	422312 pixels	515972 pixels
画像 B: テンプレートマ ッチングにより 検出された領域			
テンプレートマ ッチングにより 検出された画素 数	446331 pixels	462278 pixels	595891 pixels
画像 C: 画像 A と B の 合成画像			
正解検出領域の 画素数 (画像 C 黄)	346420 pixels	320782 pixels	474422 pixels
正解率	92.9 %	76.0 %	91.9 %
未検出領域 (画像 C 緑)	26322 pixels	101524 pixels	41550 pixels
誤検出領域 (画像 C 赤)	99911 pixels	141496 pixels	121469 pixels
不正解領域の 画素数 (画像 C 赤+緑)	126233 pixels	243020 pixels	163019 pixels
不正解率	33.9 %	57.5 %	31.6 %