

特徴点検出アルゴリズムを用いた画像によるひずみ分布解析

Image-based strain analysis using feature point detection algorithms

キーワード

ひずみ解析, 画像処理, 特徴点検出

黒田 千歳*, 歌川 紀之*, 春山 智也**,
佐藤 賢一**, 内田 崇彦**, 野原 悟**

研究概要

特徴点検出アルゴリズムを用いた画像によるひずみ分布解析の方法を開発した。この方法をいくつかの
載荷試験に適用し、試験体の状態を反映してひずみの分布の相違が異なる結果や、別のひずみ解析による
結果との整合性、また、測定を実施する上で気をつけなければならない問題点などを見出すことができた。

1 はじめに

コンクリートや鋼材を対象とした載荷試験において、その載荷前後における試験体の変位やひずみを検出する手法として、撮影した画像を利用し、解析する手法がある¹⁾。この手法によれば、荷重の変化による検査面全体の変位やひずみを一度に測定することが可能であり、時間変化の様子を追跡することができる。そこで、今回、ひずみを図化することを目的に、画像処理で使用する特徴点を検出するアルゴリズムを用い、画像によって得られた変位より図化することについて検討を行った。以下に、検討した手法および実際に載荷試験を行った際に撮影した画像を適用した結果について述べる。

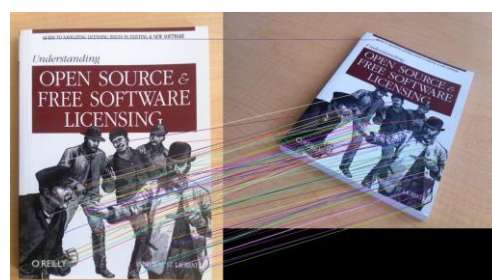


図-1 AKAZE による特徴点の対応例

2 用いた手法

2.1 特徴点検出

画像間において一致する点を見つけるために画像の各点の特徴記述子を求める手法については、Lowe によるスケール不変特徴量変換 SIFT(Scale Invariant Feature Transform)²⁾を始めとして多くのものが考案されている。特徴記述子は、画像のピクセルごとに周囲のピクセルとの勾配などから求められる多次元ベクトルであり、2画像の類似するベクトルを検索することにより、対応する点を見つけることができる。今回は、画像処理ライブラリ OpenCV で利用できる AKAZE 特徴量³⁾を用いたマッチングアルゴリズムを使用した。図-1に角度や向きが異なるように撮影された一般的な画像について、このアルゴリズムで検出された特徴点の対応例を示す。回転や縮小、射影の影響を受けずに対応点が求められていることがわかる。図-2は、同様に載荷前後の圧縮試験体ピースについて検出された

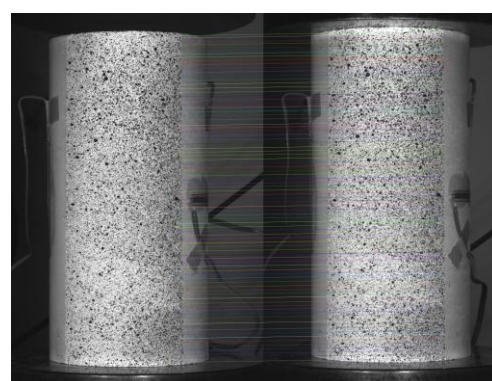


図-2 載荷前後の圧縮試験体ピースでの対応例

対応例である（見やすくするために一部の対応のみ示している）。図-3は検出されたいくつかの対応点を拡大したものである。

2画像が同じ位置から撮影されていれば、各画像での対応する特徴点の位置の変化は載荷による変位に対応する。

* 技術センター 土木研究部, ** 技術センター 建築研究部

2.2 探索された変位の選択

図-3の例のように2画像の特徴点の対応より求められた変位の一部をプロットすると図-4のようになり、変位全体は図-5のようになる。以下、変位に関するグラフの軸の単位は画像のピクセル数である。

図-4では赤い丸が変位の始点、青い線は変位の方と向きと大きさを表している。明らかに誤った対応によるものも含まれており、特徴点抽検出による誤りが発生するのを防ぐことはできない。そこで、検出された変位(図-5)を大きさの順に並び替えて累積分布を見てみると図-6のようになる。

図-7は図-6の横軸の小さい値の部分を拡大したものであるが、変位が小さい部分の密度は集中しており、この部分に対応する変位は図-8のように正しいものとなる。探索された変位が大部分正しいものであれば、一定の密度以下の変位を除外することにより正しい変位を選び出すことができる。

2.3 ひずみの計算

図-9のように、 i, j, m を頂点とする三角形要素の頂点の変位が $\{\delta_i\}^T = [u_i, u_j]$ と与えられたとき、定ひずみ要素内の変位場を、有限要素法と同様に

$$\begin{aligned} u &= \alpha_1 + \alpha_2 x + \alpha_3 y \\ v &= \beta_1 + \beta_2 x + \beta_3 y \end{aligned} \quad (1)$$

と仮定すると、 u_i, u_j, u_m を用いて

$$u = \frac{1}{2\Delta} \{ (a_i + b_i x + c_i y) u_i + (a_j + b_j x + c_j y) u_j + (a_m + b_m x + c_m y) u_m \} \quad (2)$$

ここで

$$\begin{aligned} a_i &= x_j y_m - x_m y_j \\ b_i &= y_j - y_m \\ c_i &= x_m - x_j \end{aligned} \quad (3)$$

$$2\Delta = \det \begin{bmatrix} 1 & x_i & y_i \\ 1 & x_j & y_j \\ 1 & x_m & y_m \end{bmatrix} \quad (4)$$

v も同様に与えられ、要素ひずみは、

$$\begin{aligned} \{\epsilon\} &= \begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_{xy} \end{Bmatrix} = \begin{Bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \end{Bmatrix} \\ &= \frac{1}{2\Delta} \begin{bmatrix} b_i & 0 & b_j & 0 & b_m & 0 \\ 0 & c_i & 0 & c_j & 0 & c_m \\ c_i & b_i & c_j & b_j & c_m & b_m \end{bmatrix} \{\delta\} = [B] \{\delta\} \end{aligned} \quad (5)$$

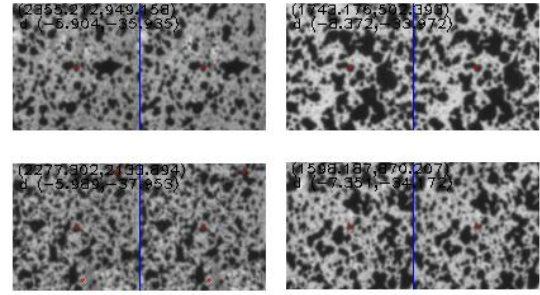


図-3 検出された対応点例

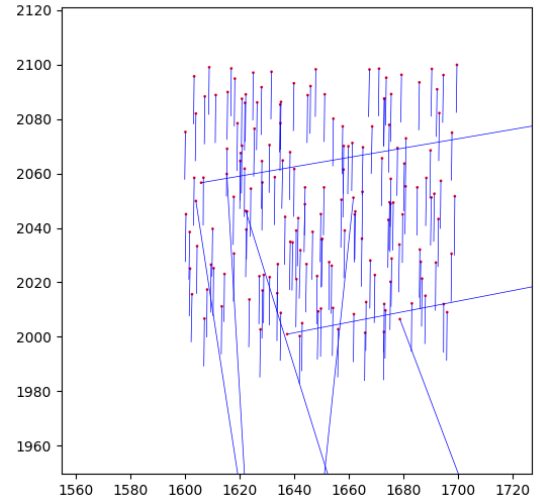


図-4 変位 (一部)

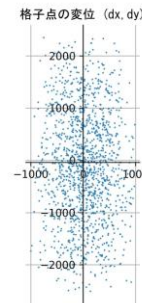


図-5 変位

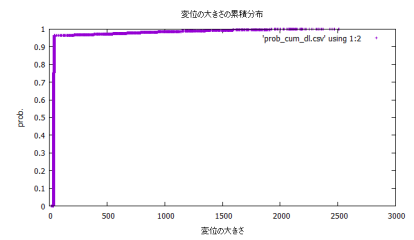


図-6 累積分布

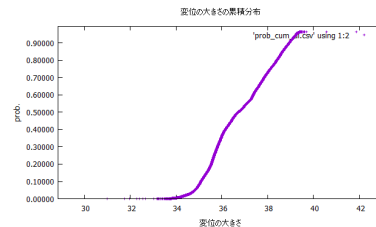


図-7 累積分布の変位が小さい部分

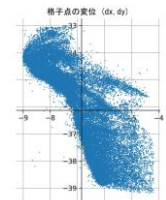


図-8 対応する変位

のように、頂点の変位より求められる。探索された特徴点は図-4で示したようにランダムな位置にある。ひずみを計算するために、この特徴点をドロネー三角分割すると図-10のようになる。

1,000個の特徴点で三角分割された要素のひずみを計算すると図-11のようになる。図は赤（圧縮）と青（引張）の濃さでひずみの大きさを表している。また、使用する特徴点を5,000個に増やして塗り分けてみると図-12のようになり、より細かなひずみの分布図が得られるが、不規則な三角形では大きさと形が不揃いで見づらく、また、非常に小さな鋭角や、鈍角を含む三角形要素によるひずみの計算の精度は低下する。

そこで、あらかじめ一定の大きさの格子点を与えて、ランダムに分布する特徴点から求めた変位を格子点の位置で補間することにする。

図-13のように特徴点が分布する場合、図-14のように各格子点のまわりにいくつか特徴点があるような間隔を設定する。

図-15は緑の矢印で示した特徴点の変位を青の格子点の位置で補間した例を示す。

変位の補間は次のように行う。

格子点 $U_{k,l}$ から距離 $r_1, r_2, r_3, \dots (r_1 < r_2 < r_3, \dots)$ の位置に特徴点 U_1, U_2, U_3, \dots があるものとする。このとき $U_{k,l}$ の変位 $U_{k,l}(d)$ を、特徴点 U_i の変位 $U_i(d)$ に距離による重みをかけて、

$$U_{k,l}(d) = \frac{\sum_{i=1}^n \frac{1}{r_i} U_i(d)}{\sum_{i=1}^n \frac{1}{r_i}} \quad (6)$$

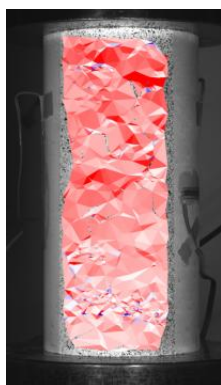


図-11 1,000 個の特徴点
によるひずみ

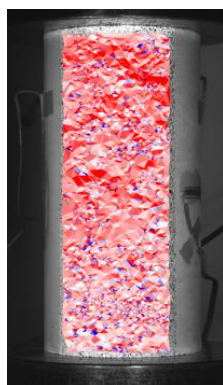


図-12 5,000 個の特徴点
によるひずみ

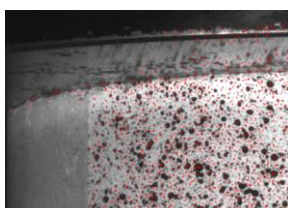


図-13 特徴点の分布

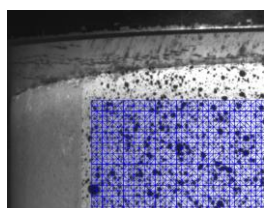


図-14 格子点の設定

とする。近傍の特徴点の数 n は数個程度とする。

図-14のように格子点に三角形要素を作り、補間された頂点の変位を用いて要素ひずみを計算すると図-16のようになった。

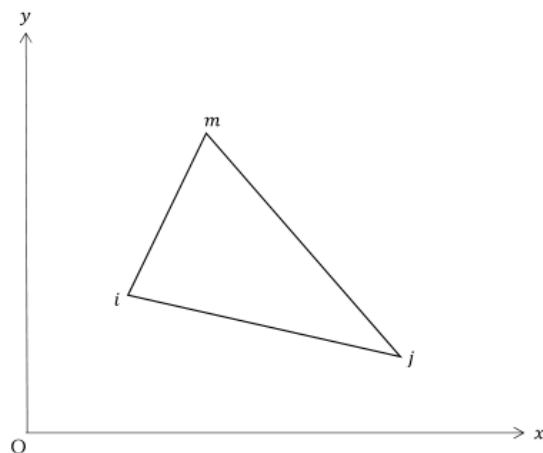


図-9 三角形要素

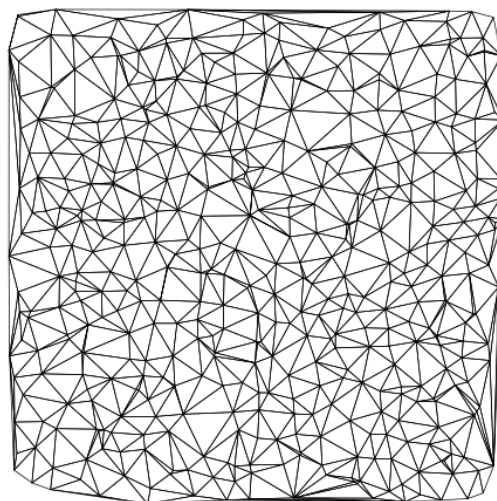


図-10 特徴点の位置のドロネー三角分割

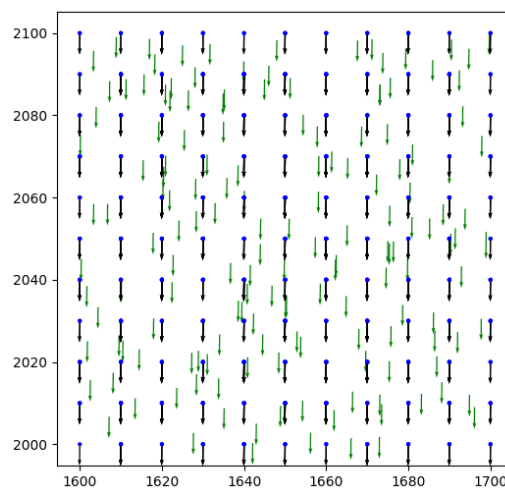


図-15 特徴点の変位の格子点位置での補間

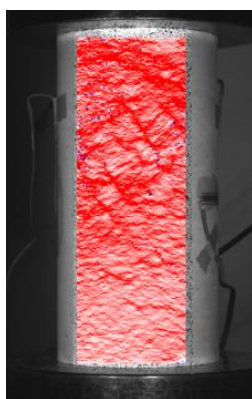


図-16 格子点の三角形要素による要素ひずみ

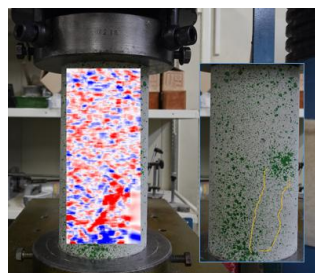
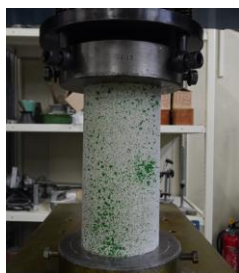


図-17 テストピース 図-18 ひずみと破断後のひび割れ位置

3 測定例

3.1 圧縮強度試験

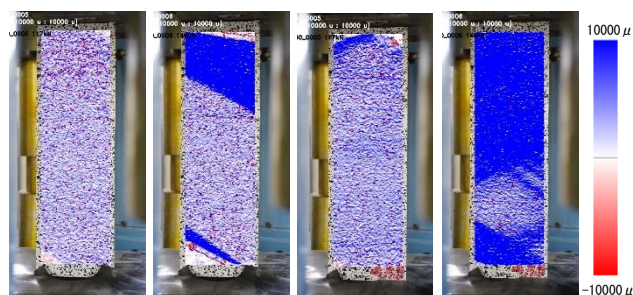
図-17のようなコンクリートのテストピースにランカーでランダムパターンをスプレーし、表面にひび割れが生じるまで圧縮を行った。図-18は破断直前のひずみと破断後にひび割れが生じた位置を示している。ひずみは赤で圧縮、青で引張を表している。テストピース下部中央の圧縮と引張が集中している箇所にひび割れが生じたことがわかる。ただし、この測定ではスプレーしたランダムパターンがかなり偏っており、右下のパターンが塗布されなかった位置には特徴点が見つからず、ひずみも正確に計算できなかった。

3.2 鋼材の引張試験

板厚4.5mmの鋼材(SS400)にランダムパターンをスプレーし、鋼材が降伏するまで引張を行った。図-19は鋼材表面の黒皮を除去しておらず、図-20は黒皮を除去したうえでランダムパターンをスプレーしている。

引張中に、鋼材が一定の変形量に達する毎に撮影を行っており、図-19,20ともに鋼材全体の縦ひずみが(a)降伏前は $1,340\mu\epsilon$ 、(b)降伏後は $1,670\mu\epsilon$ 時点で撮影している。

黒皮の有無に関わらず、降伏前までは概ね一様にひずみが分布しているのに対し、黒皮ありの図-19は、降伏後にひずみが大きい部分と小さい部分で明確に分かれている。これは、鋼材表面の黒皮が鋼材の変形に追従せず、破断するように変形したためであると考えられる。



(a)降伏前 (b)降伏後

図-19 黒皮あり

(a)降伏前 (b)降伏後

図-20 黒皮なし

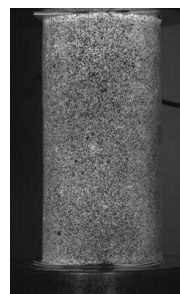


図-21 円柱コア

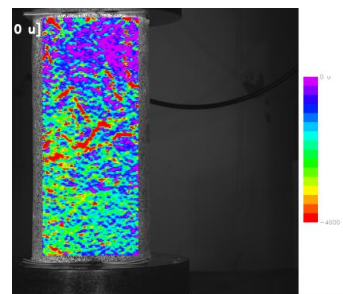


図-22 本手法による結果

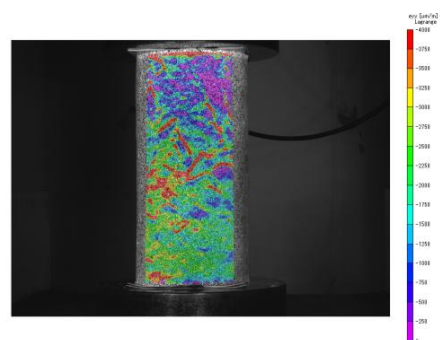


図-23 VIC-3Dによる結果

3.3 火害コンクリートの圧縮強度試験

被災したコンクリート部材を模して、片側を燃焼したコンクリート板より採取された円柱コアの圧縮強度試験の画像(図-21)をご提供いただき、これを用いて本手法によりひずみの計算を行った結果を図-22に示す。

図-21の画像は市販のひずみ計測システム「VIC-3D」で撮影されたものである。この画像から本手法を用いて計算された図-22のひずみは、最大荷重256.9kNのときのものである。コアの上側が燃焼を受けており、一様でないひずみが生じているものと考えられる。図-23がVIC-3Dによるひずみの解析結果であり、図-22ではひずみの塗り分けをVIC-3Dと同様に行っている。VIC-3Dによる結果と類似の画像が得られている。

4 ひずみの解析のために用いた手法について

土木や建築で載荷試験の画像よりひずみを検出するために変位を求める手法としては文献1)をはじめ、一般に、画像を一定の大きさの矩形に分割して、対応する位置と輝

度値パターンの相関が最大となる移動後の位置を探索するデジタル画像相関法（DIC : Digital Image Correlation）と呼ばれる手法が多く用いられているようである。今回特徴点の変位の検出に用いた手法はこれとは異なり、流れの可視化などで用いられる粒子追跡法⁴⁾に近いものである。

5 おわりに

今回、特徴点検出のアルゴリズムを用いて得られる変位よりひずみを図化することを試みたが、ひずみは変位の勾配から求められる値であり、変位の測定には高い精度が要され、使用するカメラのレンズや照明を含めた画像の撮影方法やスペックルパターンの塗布の均一性などが解析の精度に大きく影響するものと思われる。今後はこれらのことも考慮して実験を行い、また、载荷試験に合わせた FEM のモデルを作成して、画像によるひずみの解析結果の検証を行う予定である。

謝辞

本報の3.3節で示した実験データは、日本建築総合試験所と富山県立大学の内田慎哉先生により行った共同研究の成果の一部をご提供いただいたものです。内田慎哉先生には貴重な機会をご提供いただきましたこと、ならびに日本建築試験所には本報で使用した実験データをご提供いただきましたことに、感謝いたします。

参考文献

- 1) 車谷麻緒, 松浦遵, 根本忍, 呉智深: コンクリートのひび割れ進展計測のための画像解析手法に関する基礎的研究, 土木学会論文集 A2 (応用力学), Vol.70 No.2 (応用力学論文集 Vol.17), I_135-I_144, 2014.
- 2) D.G.Lowe: Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision, #60, 91-110, 2004.
- 3) OpenCV 3 と Python 3 で特徴量マッチング (A-KAZE, KNN) <http://www.robosafe.com/personal/pablo.alcantarilla/kaze.html>
- 4) 可視化情報学会: PIV ハンドブック (第 2 版), 森北出版, 2018.