

3次元画像からの空隙測定方法の開発

大平 倫宏*¹⁾ 富山 真一*²⁾

Development of a void measuring method from 3D images

Norihiro Ohira*¹⁾, Shinichi Tomiyama*²⁾

Voids in materials and industrial products are important from the perspective of quality management and performance management. In this paper, we developed a non-destructive image processing method that can measure voids using X-ray computed tomography equipment. A newly-developed 3D labeling technique with a memory saving and high speed processing performance was used. In the experiment, we took measurements of the voids in the solder on circuit boards, confirming that it is possible to calculate the voids using one-twentieth of the memory space compared with the conventional method.

キーワード: X線 CT, 空隙, ラベリング, 画像処理

Keywords: X-ray computed tomography, Void, Labeling, Image processing

1. はじめに

工業用素材全般の開発・生産において, その素材の空隙が重要となることがある。特に, 電子製品のはんだ中の空隙は, 品質管理上重要視され, 大きな空隙が存在する場合や全体に占める空隙の割合が大きくなった場合, 製品寿命の低下を招くことが知られている。

素材の内部構造を確認するために用いられる方法は, 破壊検査と非破壊検査に分類できる。破壊検査では, 素材を切断するなどの非可逆操作を行い, その断面を見るなどの方法が存在する。しかし, 破壊検査では, 素材によっては切断中の衝撃によって空隙が変化してしまい, 正しいデータが得られないなどの問題が生じる。一方, 非破壊検査では, X線 CT (Computed Tomography: コンピュータ断層撮影) 装置や超音波 CT 装置が利用されている。超音波 CT は, 対象が水分を多く含んでいる場合は有効であるが, 一般的に水分をあまり含まない工業用素材に対しては正しいデータが得られない。そこで, X線 CT 装置を用いた非破壊検査で空隙を測定することが求められている。

本研究では, 電子機器のはんだの空隙を測定する方法の開発を行った。図 1 の電子基板のはんだ中の空隙を X線 CT 装置を用いて撮像した断層画像を図 2 に示す。空隙部分が黒く鮮明に映されている。このように, 電子基板のはんだ中の空隙については, 比較的鮮明な画像が得られる傾向がある。このため, 従来よく使用されている方法である 2 値化処理⁽¹⁾とラベリング処理によって空隙の計数, 個々の空隙の大きさの測定, 全体に占める空隙の割合の測定が可能

であることが期待される。ここで, ラベリング処理とは, 図 3 のように, 2 値画像に対して連結している部分を同じグループとして識別し, 個々のグループに異なるラベルを付ける処理を言う。

X線 CT 装置を用いて空隙を発見するためには, 複数の断層画像についてラベリング処理を行う必要がある。このため, 計算量とメモリ消費量が多い。そこで, 本研究では Graphics Processing Unit (以下, GPU) を用いたラベリング処理の並列実行⁽²⁾によって処理時間の短縮を図る。一方, GPU を用いた並列処理では, Central Processing Unit (以下, CPU) を用いた場合に比べて利用できるメモリ容量が少ない。X線 CT 画像を対象としたラベリング処理ではメモリが不足する可能性がある。そこで本研究では, GPU を用いた並列計算が可能であり, 消費メモリ量が従来に比べて少ないラベリング手法を開発した。

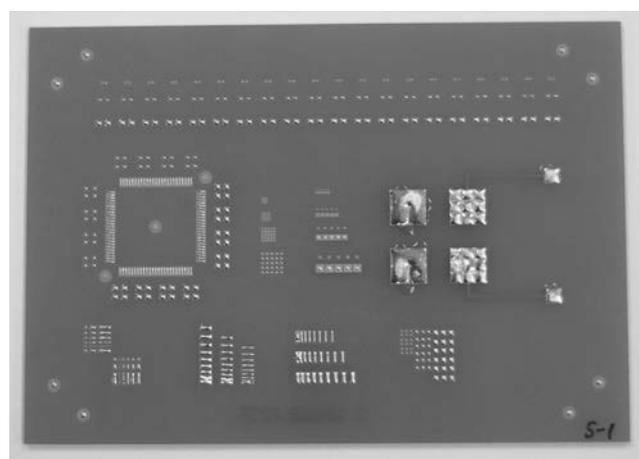


図 1. はんだ付けされた電子基板

事業名 平成 27 年度 基盤研究 「3次元画像からの空隙率測定方法の開発」

*1) 生活技術開発セクター

*2) 情報技術グループ

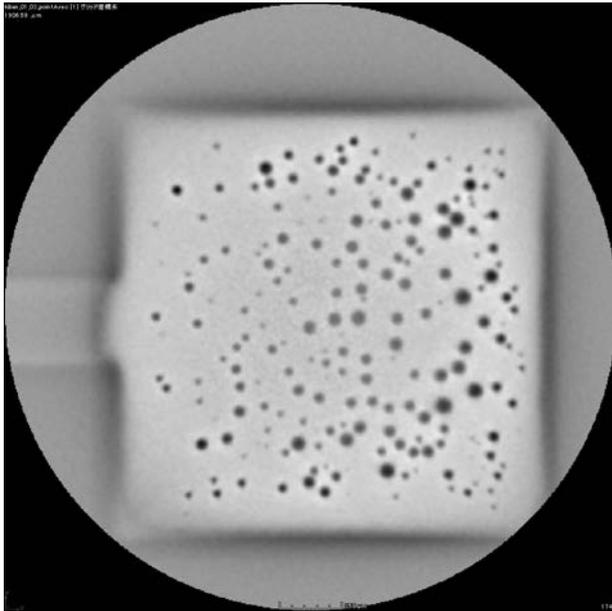


図2. 図1の基板の断層画像

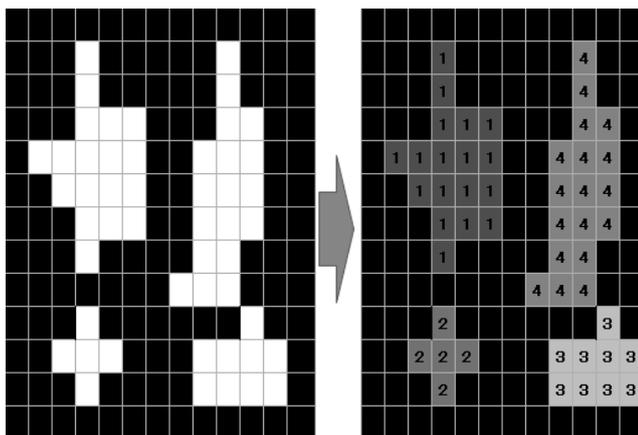


図3. 2次元画像に対するラベリングの例

2. 開発した方法

2.1 処理手順 開発手法の処理の流れは以下のようになる。

- (1) 2値化 得られた断層画像のすべてを対象として、空隙部分と背景の2種類に分けるため、2値化処理を行う。
- (2) ラベリング 2値化した画像に対して、以下に述べる3次元ラベリング処理を行い、空隙毎にグループ分けを行う。
- (3) 空隙の測定 空隙のグループ数を計数することや、グループ毎の体積を求めることで、空隙の測定を行う。

2.2 3次元ラベリング方法 2次元画像を対象とした逐次ラベリングアルゴリズムで、メモリ消費量の削減を目的とした方法として、Stripe-based⁽³⁾法がある。Stripe-based法では、最初に1方向の直線状に隣接関係を走査しながら、記憶領域を圧縮する処理を行い、以後はその圧縮された記

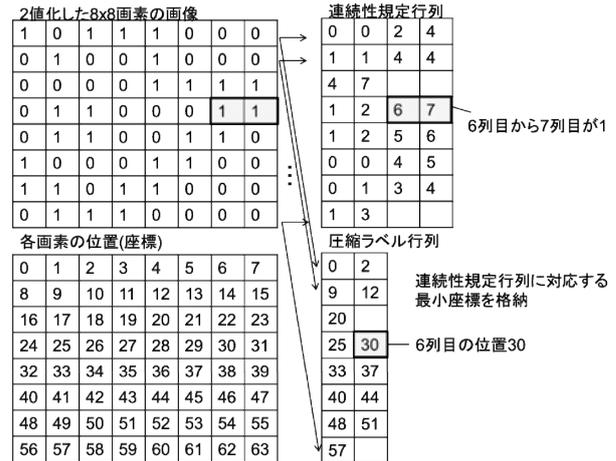


図4. 連続性規定行列・圧縮ラベル行列の計算例

憶領域を利用して処理を続行する。

本研究では、2次元画像用のStripe-based法を基に、3次元画像に対して省メモリかつ並列にラベリングを行う方法を開発した。以下では、特に断りのない限りGPUを用いた並列計算を扱うが、CPUだけを利用して同様の並列計算を行うことも可能である。

実際のラベリング処理は以下の手順に従って行う。簡単のためそれぞれの断層画像は8×8画素の画像からなるとしている。

(1) 連続性規定行列・圧縮ラベル行列の計算 すべての断層画像に対して、図4のように各行を走査して、連続性規定行列と圧縮ラベル行列を作成する。

図4左上は2値化した画像であり、0が背景、1が空隙を表している。図4左下は、各画素に対応する3次元空間上の位置である。図4右上は、同図左上の2値化画像の各行について、連続した空隙を開始列番号、終了列番号の順で順次記憶した連続性規定行列である。また、図4右下は同様に2値化画像から計算する圧縮ラベル行列であり、各行に対して連続性規定行列に対応する最小の位置を順次記憶する。各行に対して、圧縮ラベル行列の列数は、連続性規定行列の列数の半分となる。

図では、例として、3行目に対する処理を示している。まず、3行1~2列目に空隙が連続するため、連続性規定行列の3行0列、1列にそれぞれ1、2を記録する。同様に2列、3列にそれぞれ6、7を記録する。また、圧縮ラベル行列に対しては、空隙の連続の開始位置(座標)である25、30を順次記録する。

各行の処理は、他の行の処理に関係なく独立して並列計算可能である。また、GPUメモリ上には、その都度に必要な部分の画像データだけを転送して、メモリ使用量を削減することが可能である。

(2) 隣接関係の計算 (1)で計算した連続性規定行列と圧縮ラベル行列から、各行に対して、図5のように隣接関

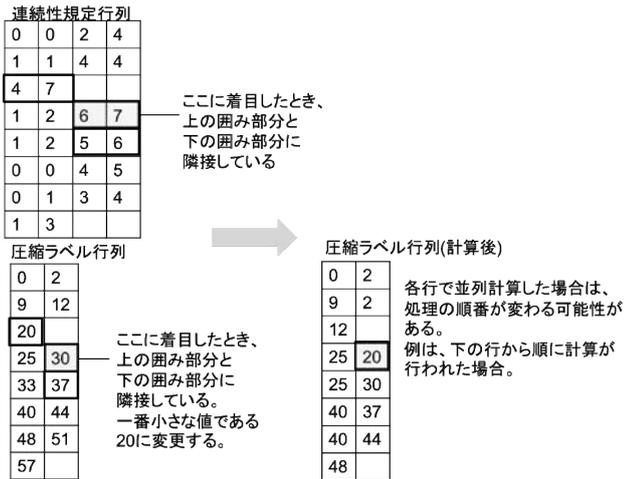


図 5. 隣接関係の計算例

係の計算を行う。

例として、連続性規定行列の 3 行 2 列目から 3 行 3 列目の値に着目した場合の処理を示す。この値は、2 値画像の 3 行 6 列から 7 列に空隙が連続することを意味している。次に連続性規定行列の着目行に隣接する 2 行について、連続する空隙の位置を探索する。2 行 0 列目が 4、2 行 1 列目が 7 であるため、開始列、終了列の区間は[4, 7]となり、着目区間[6, 7]と重なっているため、元の断層画像において隣接している。その際、隣接部分に対応する圧縮ラベル行列の値は 20 である。ここで 2 区間の重なり判定は、2 区間を $[s1, e1]$, $[s2, e2]$ と表した時、 $s1 \leq e2$ かつ $e1 \geq s2$ である場合に、2 区間が重なっていると判定する。

同様に、一つ下の行である 4 行目については、1 つめの区間は[1, 2]となり、着目区間[6, 7]とは重なっていない。次に、2 つめの区間[5, 6]は、着目区間[6, 7]と重なっており、元の断層画像において、隣接している。ここでは、隣接部分に対応する圧縮ラベル行列の値は 37 である。

次に、隣接部分が存在する場合、着目部分に対応する圧縮ラベル行列の値を、隣接部分および着目部分に対応する圧縮ラベル行列の値のうちで最小の値に更新する。例の場合では、着目部分に対応する圧縮ラベル行列の値は 30 であり、隣接している圧縮ラベル行列の値が 20 と 37 であるため、着目部分に対応する圧縮ラベル行列の値を 20 に更新する。

図 5 では、簡単のため 2 次元画像の着目行列に対して、同じ画像の 1 行前と 1 行後だけを参照しているが、実際には 3 次元の隣接関係を計算するため、3 次元的に隣接する断層画像の同じ行に対しても、同様に隣接関係の計算を行う必要がある。その場合でも、各行の処理は他の行と独立に並列計算可能である。

これらの手順を圧縮ラベル行列に変化がなくなるまで繰り返して、隣接関係の計算を終了する。圧縮ラベル行列は上の行ほど小さな値を持つため、隣接関係の計算は上の行から行われることが望ましい。しかし非同期の並列処理で

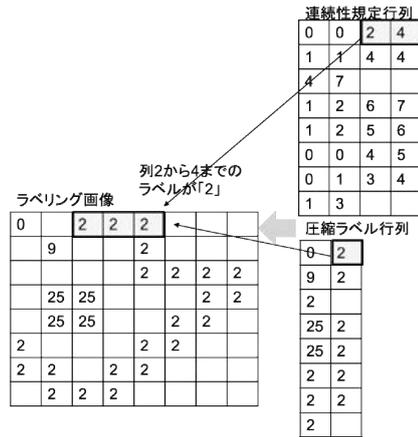


図 6. ラベリング画像の計算例

は、並列に実行されるそれぞれの処理がどのような順番で実行されるかを規定することはできない。図 5 右下は、下の行の処理が先に行われた最悪の場合を表している。このような場合でも、早期に計算を収束させるための手法として、Find Root 処理^{(2), (3)}が存在するが、ここでは詳細については割愛する。Find Root 処理は必ずしも行う必要はないが、今回の開発では、実装を行った。

(3) ラベリング画像の計算 (2) で計算した圧縮ラベル行列から、最終結果であるラベリング画像の計算を行う。基本的には(1)と逆の処理であり、各行に対して、連続性規定行列から空隙の開始列番号と終了列番号を、圧縮ラベル行列から対応する空隙のグループを取得して、それらの情報を基に、ラベリング画像を作成する。

図 6 の例では、連続性規定行列の 0 行 2 列目から 0 行 3 列目の値と圧縮ラベル行列の 0 行 1 列目の値に着目した場合の処理を示している。連続性規定行列では、列の 2 番目から 4 番目までが空隙であり、圧縮ラベル行列では、その時のラベルのグループが 2 であることを示しており、それらに応じてラベリング画像に反映している。

2.3 結果 株式会社ユニハイトシステム製の XVA-160

α を用いて取得した 3 次元断層画像について提案法を適用した。断層画像は、図 2 のような縦横 1000×1000 ピクセルのグレースケール画像とし、1 回の撮像で 600 枚を取得した。この中で、はんだ中の空隙が映っている 100 枚ほどを対象として、Label Equivalence 法⁽²⁾と提案手法の計算時間、消費メモリ量を比較した(表 1 参照)。実行環境は、

- OS: Windows 7
- CPU: Core i7-3770K 3.5GHz
- ホストメモリ: 16 GB
- GPU ボード: STRIX-GTX980-DC2OC-4GD5 (GPU メモリ 4 GB)
- プログラム開発環境: Visual Studio 2013
- プログラム言語: C++, CUDA7.5

である。

究や開発支援事業での利用が可能である。

(平成 28 年 7 月 5 日受付, 平成 28 年 8 月 8 日再受付)

文 献

- (1) 大津 展之: 「判別および最小 2 乗規準に基づく自動しきい値選定法」, 電子情報通信学会論文誌, Vol.J63-D, No.4, pp.349-356 (1980)
- (2) K. A. Hawick, A. Leist, D. P. Playne: “Parallel Graph Component Labelling with GPUs and CUDA”, Parallel Computing, Vol.36, No.12, pp.655-678 (2010)
- (3) H. L. Zhao, Y. B. Fan, T. X. Zhang and H. S. Sang: “Stripe-based Connected Components Labelling”, Electronics Letters, Vol.46, No. 21, pp.1434-1436 (2010).

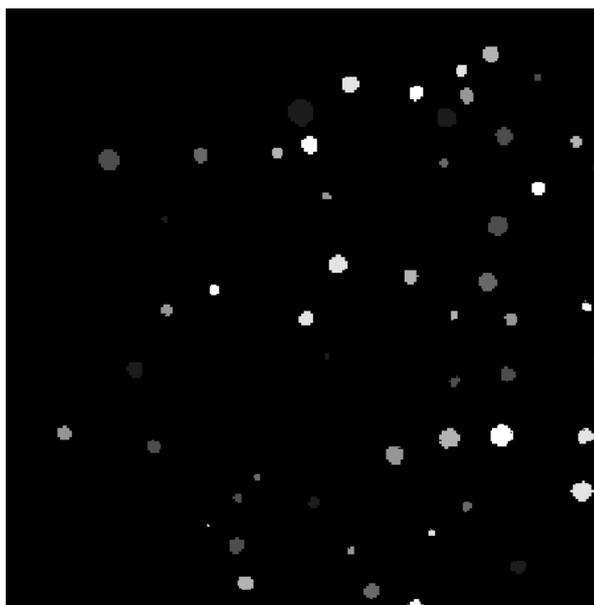


図 7. ラベリング画像(図 2 対応部分の一部)

表 1. ラベリング方法の比較

手法	処 理 時 間 (msec)	消 費 メ モ リ (GB)
Label Equivalence 法 ⁽²⁾	778	約 0.8
提案法 (メモリ 削減なし)	25	約 0.53
提案法 (メモリ 削減)	315	約 0.04

提案法では Find Root 処理を利用した。また, 処理時間は PC メモリ上に画像を読み込んだ後から計測を開始し, GPU における計算時間, GPU と PC メモリ間のメモリ転送時間を含む。

表 1 から, 提案法が従来法と比べて, GPU 上の消費メモリ量が少ないことがわかる。特に, 都度部分画像データを転送するようにした場合, PC と GPU メモリ間のデータ転送を繰り返すことによって処理時間は長くなるが, GPU の消費メモリ量は大きく削減される。

図 7 は, 実際に, 図 2 を含む 3 次元断層画像に対して提案法を適用した結果の一部であり, 各ラベルに対して異なった濃淡をつけている。

3. まとめ

X 線 CT から得られる 3 次元断層画像に対して, その空隙を計測する方法について, ラベリング処理を用いる方法を新たに開発した。データ量の多い 3 次元断層画像に対応するため, 従来の方法に比べて, ボトルネックとなり得るメモリ使用量を減少させる並列ラベリング方法を考案した。提案法では, ラベリングを行うことで, 空隙の計数や各空隙の大きさの計測, 全体における空隙の割合の計測等が可能となる。

今回開発した方法については特許出願中である。共同研