

5 画像処理技術を活用した自動車部品等の表面状態高速検査技術の開発 (第3報)

弓場憲生, 佐野 誠, 石津任章, 打田澄雄

Development of rapid image processing inspection for state on surface about some of automotive parts (3rd Report)

YUBA Norio, SANO Makoto, ISHIZU Hideaki and UCHIDA Sumio

Automotive parts are tested by visual inspection in many factories. However, it is unstable in measuring a fixed quantity. Development of automatic products inspection system using image processing is highly hoped.

The products of the research are as follows: 1) Software library for image processing was developed. This library was made by OpenCV, Python and SWIG 2) Many types of intellectual properties for image processing were made. Some of them and cameras were connected with local area network system. 3) Defective plastic parts were detected by RT method of MT system in curvature judgment software.

キーワード:画像処理 IP, ライブラリ, プログラム, 欠陥検査

1 緒 言

自動車部品等の製造業において、高品質な製品を安定生産するためには欠陥検査が不可欠であるが、その行程ははまだ職人的な技能に頼っているのが実態である。そこで本研究では、検査工程の自動化、省力化を目標に、①画像処理技術を円滑に導入・活用するための知識データベースの開発（モデルドリブン画像処理設計支援技術の開発）、②所定時間内に欠陥を検出する画像処理 IP(Intellectual Properties, 知的財産)技術の開発（高速高精細画像処理用 IP の開発）、③各種検査対象ごとに最適化されたソフトウェアを搭載した欠陥検出装置を開発（欠陥検出技術の開発）した。

2 画像処理設計支援基盤の構築

2.1 研究の目的

当センターではデータ量の多い高精細画像を高速で処理する IP の開発を行っており、企業の製品開発や試作品開発に利用されている¹⁾。これらは回路の小型化、高速化を可能にするため、特に組み込み開発の用途に向いている。さらにこの IP と同等の機能を有するソフトウェアライブラリも開発しており、ソフトウェア上でシミュレーションが可能である^{1) 2) 3)}。

本研究では、より効率的に画像処理技術を応用した製品開発等を支援するため、汎用画像処理ライブラリ

とインタプリタ型言語の採用、インタプリタ型言語の機能拡張を行う技術を確立した。インタプリタ型言語は処理速度は遅いがプログラムを簡単に記述し、コンパイルせずに実行できるためアルゴリズムの検証等に有効である。高速性が要求される場合には、インタプリタ型言語でプログラム全体を作成し、動作が検証された部分のコードを C/C++で作成する。さらに高速化が必要であればハードウェアで処理を行う。図1に画像処理システムの設計を支援する基盤の構成を示す。

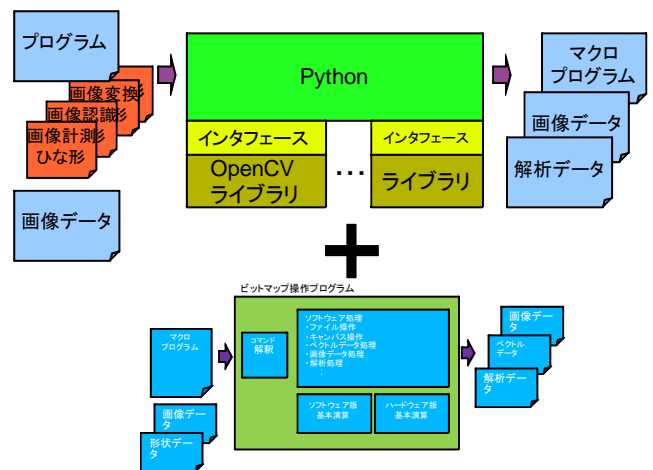


図1 画像処理システムの設計を支援する基盤構成

2.2 汎用画像処理ライブラリの概要

画像処理は画像の入力に始まり、画像の補正／解析

／計測・認識等の処理，処理結果の出力までさまざまな処理が必要となる。目的とする画像処理アルゴリズムや入出力仕様を効率良く開発するためには，既存の画像処理ライブラリを活用することが有効である。そこでオープンソースの汎用画像処理ライブラリ OpenCV を採用した。これには描画関数，画像フィルタリング，画像の幾何学変換，ヒストグラム，特徴検出，モーション解析，構造解析，平面細分割，物体検出，カメラキャリブレーション，ユーザインタフェース，画像・ビデオの読み込みや書き込み等多くの機能がある⁴⁾。主な機能を表1に示す。

2.3 インタプリタ型言語の概要

インタプリタ型言語は，コンピュータで直接実行できる形式に変換するコンパイラ型言語と比べ，プログラムを簡単に記述し実行できるので開発の効率が良い。Python は高度なプログラミング手法や大規模開発をサポートするための機能を備えながら，テキスト処理から GUI，インターネットプロトコルなど豊富なライブラリが用意されているインタプリタ型言語である。Python には，インタプリタ型，オブジェクト指向言語，豊富なデータ型，例外処理，マルチスレッド対応，豊富な拡張ライブラリ，C/C++ による拡張が簡単，等の特徴がある⁵⁾。主な特徴を表2に示す。

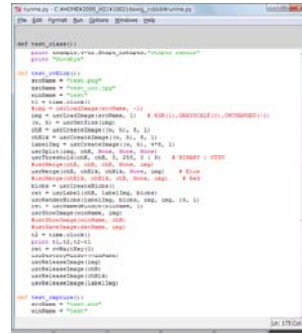
描画関数 画像フィルタリング 画像幾何学変換 ヒストグラム 特徴検出 モーション解析 構造解析 平面細分割 物体検出 カメラキャリブレーション ユーザインタフェース 画像とビデオの読み込み／書き込み
--

オープンソース インタプリタ型 オブジェクト指向言語 豊富なデータ型 (リスト等) 例外処理 マルチスレッド対応 豊富な拡張ライブラリ C/C++ による高い拡張性

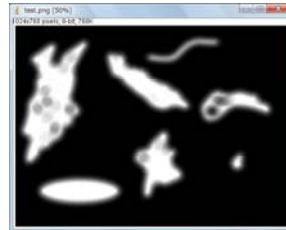
表1 OpenCV の主な機能 表2 Python の主な特徴

2.4 インタプリタ型言語の機能拡張

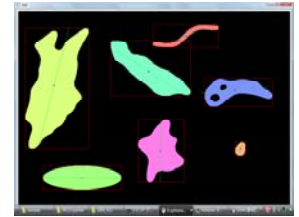
Python から OpenCV を利用するため，SWIG (Simplified Wrapper and Interface Generator) を用いた。SWIG もオープンソースであり，プログラムとインタプリタ型言語間の接続を最小限の作業で実現することができる。SWIG は C/C++ で記述されたプログラムと Python をリンクして共有ライブラリを生成する。図2は，C++で記述したラベリング機能⁶⁾を Python から利用できるようにした例である。



(a)プログラム例



(b)元画像



(c)処理結果画像

図2 ラベリング機能を Python から利用する例

3.高速・高精細画像処理用 IP の開発

3.1 概要

既に，画像処理 IP の高速化 (CameraLink 規格の最高速度 85MHz への対応) と高精細化 (8192×8192 画素) を行ったが，今年度は実検査に応用するために，画像処理 IP を実装した2種類の装置を開発した。1つは，複数台のカメラで撮像した画像を高速処理するための分散処理装置、もう1つは，カメラ，照明，分散処理装置の制御，傷検査アルゴリズムの実行，欠陥判定等を行うデータ処理装置である。これらを利用することで傷検査装置の短期開発，性能検証，低コスト化が実現できるため，県内企業が効率的に検査システムを開発することが可能となる。

3.2 分散処理装置の開発

大きなワークを複数カメラで分割して撮像する場合や，解像度の異なるカメラで1つまたは複数のワークを撮像する場合など，複数カメラによる画像処理が必要な場合に，PC もしくは専用機でシステムを構成すると，規模の増大，過剰スペック，コスト増，処理時間の増加などの問題が起こる。これらを解決するために，1台のカメラに対する処理機能を持つ分散処理装置の開発を行った。

図3は分散処理装置の一例(図4の分散処理装置①)であり，(株)アットマークテクノのSUZAKU ボードに接続した AV ボードに NTSC カメラを接続し，FPGA で取得画像から GrayScale 画像，2値画像，エッジ画像を生成している。その他にも，任意の画像処理機能を追加可能で，要求仕様に応じてハードウェア処理とソフトウェア処理の何れの実

装も可能である。また、複数構成も可能であり、トリガ制御によって同時処理やインターリーブ処理も可能である。

高解像度カメラもしくは高速処理が必要な場合は、Xilinx の Spartan3AN ボード+CameraLink 入力ボード(図4の分散処理装置②)の構成が有効である。モニタ出力もあり、単体でも簡易な検査装置が構築可能である。

これらの分散処理装置を活用することで、1台のカメラに対して必要な機能のみを実装できるため、処理時間が増加することなく、複数カメラを用いた検査装置を安価に構成することができる。

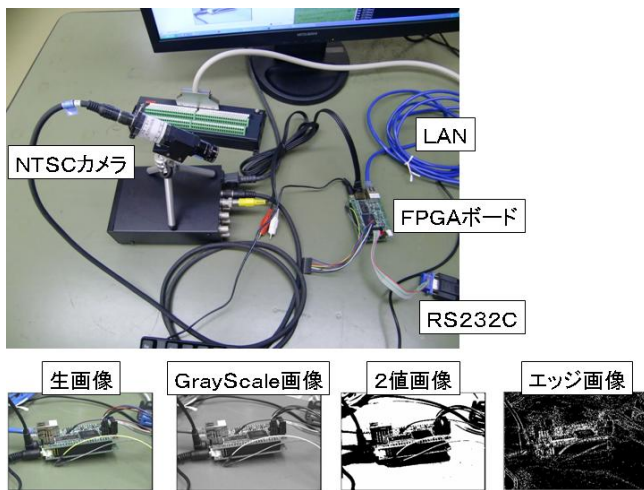


図3 分散処理装置

3.3 データ処理装置の開発

本研究で開発した設計資産(モデルドリブン, 傷検査アルゴリズム, 画像処理ライブラリ, 高速・高精細画像処理IP)を実行するプラットフォームとして、データ処理装置を開発した。主な構成は、CPU(Intel 社の Core2Duo), メインメモリ(4GB), FPGA ボード(GiDEL 社の PROCellIII-80E-B), デジタルI/Oであり、CPUとメインメモリでのソフトウェア処理(モデルドリブン, 傷検査アルゴリズム, 画像処理ライブラリ)と FPGA ボードでのハードウェア処理(高速・高精細画像処理 IP)が可能である。ハードウェア処理はCPUから画像処理ライブラリとして利用できる。また、FPGA ボードはカメラから直接取得した画像に対する処理も可能で、インライン検査に要求される高速処理にも対応できる。

図4は複数カメラを用いたときの構成例である。データ処理装置と分散処理装置①, ②には、それぞれ200万画素, 30万画素, 130万画素のカメラが接続されており、データ処理装置と分散処理装置はRS232CもしくはLANで接続されている。分散処理装置は、カメラの画像に対する処理結果をデータ処理装置に転送するため、データ処理装置で直接処理するよりも処理時間が短縮できる。しかし、

高解像度カメラへは対応できないため、高解像度カメラが必要な場合は、データ処理装置で直接処理を行う。

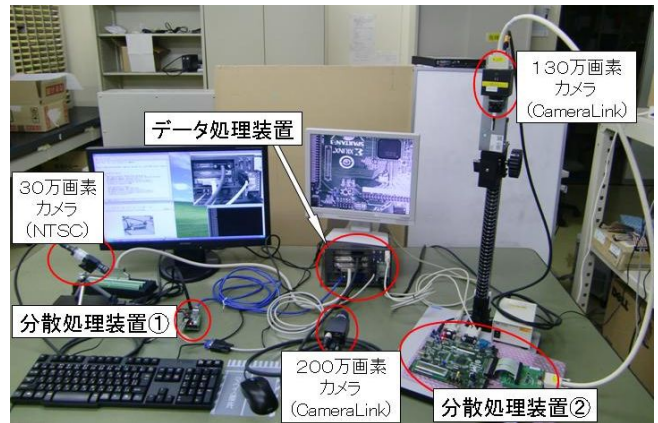


図4 データ処理装置の構成例

4 成果展開

4.1 プラスチック成形品の検査

本研究では様々な製品の欠陥検出を行ったが、ここではプラスチック成形品の欠陥検査の事例について述べる。プラスチック成形品では、プラスチック材料や金型の温度, 射出時の速度や圧力などの条件設定が適切でないと、種々の欠陥が発生する。ショートショットは、熔融プラスチックが金型キャビティ(成形品の形状となる金型内部の空間)の隅々にまで行き渡らず、キャビティを完全に充填しきれない欠陥である。



図5 検査対象のワーク(左:不良品, 右:良品)

図5は検査対象のワークである。プラスチックが金型に十分に充填していないので、良品と比較して輪郭が不均一になる。そこで、曲率を使ってワークの輪郭形状を表現し欠陥検出を試みた。欠陥の判定にはMTシステムのRT法を採用した。

4.2 画像処理の概要

画像処理の手順について説明する。

- 1) 撮影したワークの画像を2値化し、輪郭線を求める
(本研究では8近傍)
 - 2) 輪郭内部の物体領域から重心、短軸を求め、短軸と輪郭線の交点のうち上部側を開始点とし、点Oを設定する
 - 3) 点Oから左右に距離d離れた点P, Qを求める
 - 4) 3点O, P, Qから曲率Cを計算する
 - 5) 輪郭に沿って、点Oを移動量mだけ反時計回りに移動させる
 - 6) (3)~(5)を繰り返しながら点Oを1周させる
- 曲率Cは、3点O, P, Qの座標を円の方程式に代入して曲率半径を求め、この逆数で定義する。

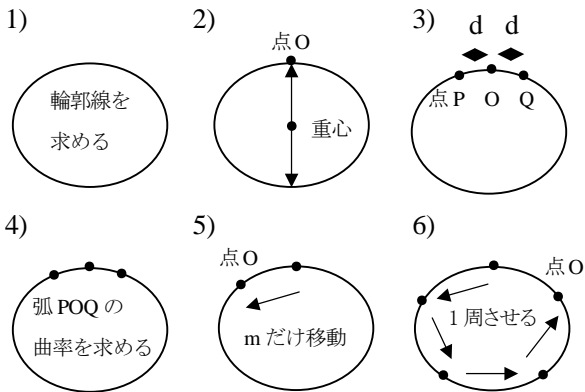


図6 画像処理の手順

4.3 実験結果

実験では640×480画素でワークを撮影した。良品サンプルは10個(No.1~No.10)、不良品サンプルは5個(No.11~No.15)用意した。ワークの重心から反時計回りに角度2°毎に輪郭を分割して(180ブロック)、各ブロック内の曲率を平均化した。良品(No.1)と不良品(No.11)の曲率の結果を図7に示す。縦軸が曲率(の平均値)、横軸がブロック番号である。

図8にRT法を用いて判別した結果を示す。横軸はサンプル番号で、1~10番が良品、11~15番が不良品のサンプルを示す。RT法とは、良品サンプル集合を使って統計的に求めた良品らしさの尺度(本研究ではσ比と呼ぶ)により、良品、不良品を判定する方法である。σ比が大きいくほど良品から外れていることを示す。良品でσ比が一番大きいもの(No.9 σ比=1.60)と、不良品でσ比が一番小さいもの(No.13 σ比=16.9)とはおよそ10倍の差があり、本手法の高い識別能力がうかがわれる。

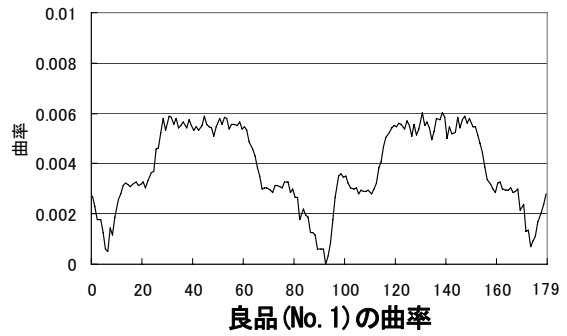
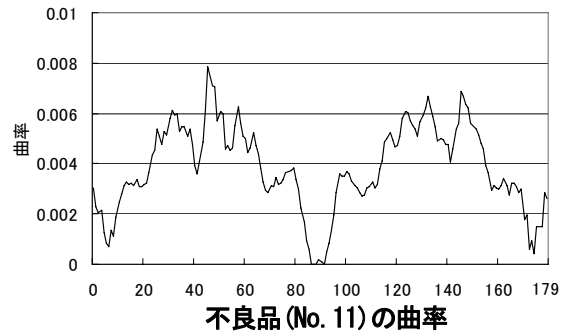


図7 輪郭上を一周して求めた曲率

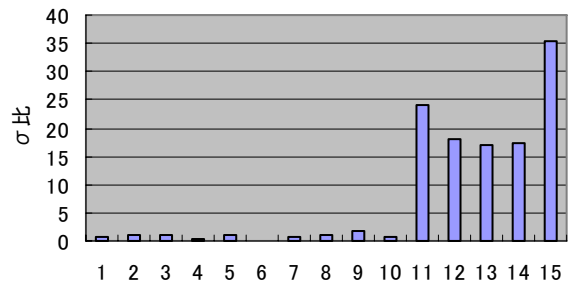


図8 良品、不良品のσ比

5 結 言

欠陥検査を自動化する画像処理プログラムを効率良く開発する画像処理ライブラリを開発した。さらにこのプログラムを高速で実行する画像処理IPを開発し、大容量画像データを高速に処理するために、複数のIPを接続した分散処理装置も開発した。また本研究で行った多くの企業対応のうち、プラスチック部品の欠陥において、曲率のバラツキをRT法で評価した結果、高い欠陥検出能力が認められた。

文 献

- 1)佐野他:広島西部工技研究報告, 51(2008), 5
- 2)佐野他:広島西部工技研究報告, 49(2006), 1
- 3)佐野他:広島西部工技研究報告, 52(2009), 36
- 4) OpenCV 2.0 (r2873) Python Reference , <http://opencv.jp/opencv-2.0svn/python/>
- 5) <http://www.python.jp/Zope/intro>
- 6)<http://code.google.com/p/cvblob/>