

物体識別AIのためのバーチャル学習画像生成の検討

Virtual learning images generation for object discrimination by machine learning

梅村 良佑^{※1}

Ryosuke UMEMURA

寺西 勇仁^{※1}

Yuto TERANISHI

千田 暁慧^{※1}

Akisato CHIDA

松村 翔太^{※1}

Shota MATSUMURA

画像を用いた機械学習を行う上で必要な学習画像の収集を簡易化するために、実際の対象ワークから3Dモデルを生成し仮想空間上で画像を収集する方法について考えた。本報告では、3Dモデルの作成を行う際の自動化・カメラパラメータの調整方法について検討を行った。また、仮想空間上から得られた画像を用いて、実環境に用意した物体の画像認識を行った結果、正解率は88.67%であった。今後は精度の向上、適用アプリケーションの探索を行う。

In order to simplify the collection of learning images necessary for machine learning using images, we considered a method of generating 3D models from actual target workpieces and collecting images in a virtual space. In this report, we examine how to automate and adjust camera parameters when creating 3D models. In addition, image recognition of objects prepared in a real environment using images obtained from a virtual space resulted in a correct recognition rate of 88.67%. In the future, we will improve accuracy and explore applicable applications.

エコロジー訴求点と 顧客メリット

- ①省人化・無人化
- ②作業改善

1. はじめに

近年、AIという言葉はニュースなどでも一般的に利用されるようになってきている。このAIの分野は、処理するPCの高速化なども伴い、ディープラーニングを用いた手法が様々開発されている。物体判別の技術としては、身の周りのものにスマートフォンを向けるだけで物体の名前を表示させるような技術（Google レンズ[®]）も存在する。これらは、物体認識技術の中でも、犬・車・飛行機などグループを認識する一般物体認識と呼ばれている。

当社でも省人化・自動化のために、物流センター等に使用される仕分け装置に適応可能な物体判別装置の開発を行っている（図1）。昨年度の報告では、バーコードやRFIDを使用せずに各々商品の特定を行う、特定物体認識技術について述べた。また、この技術を落下物体の判別に応用した技術デモ装置『AIでむ鑑定士[®]』（図2）の製作を行い、問題なく判別が可能であることを報告した¹⁾。

一方で、これらの取り組みの中で、学習に必要な画像

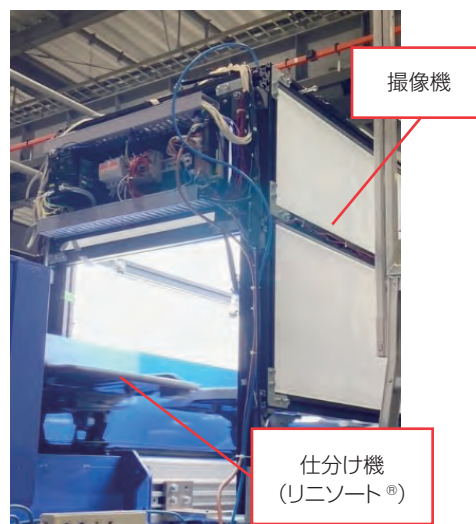


図1. 物体判別装置

を取得する負担が非常に大きいことがわかった。また、実際に仕分け装置に判別装置を適応させていく上で、対象となるワークのパッケージ変更や新商品への対応が必要であり、学習画像取得の負荷低減が必要と考えた。

そこで対象ワークの3Dモデルを作成し、バーチャル空間上（PC内部での仮想環境）に配置し画像を撮像する。これを3Dモデルの配置や姿勢を様々に変化させ、

※1 研究開発センター 先端技術部

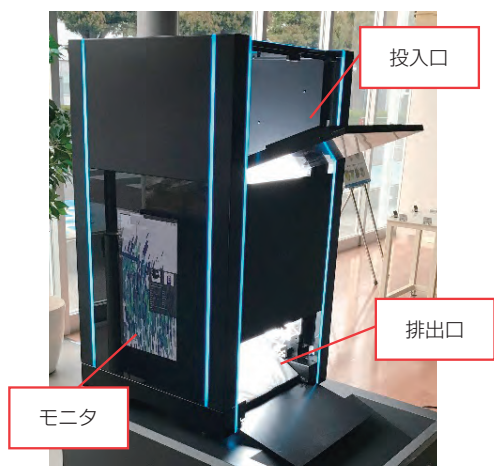


図 2. AI でむ鑑定士

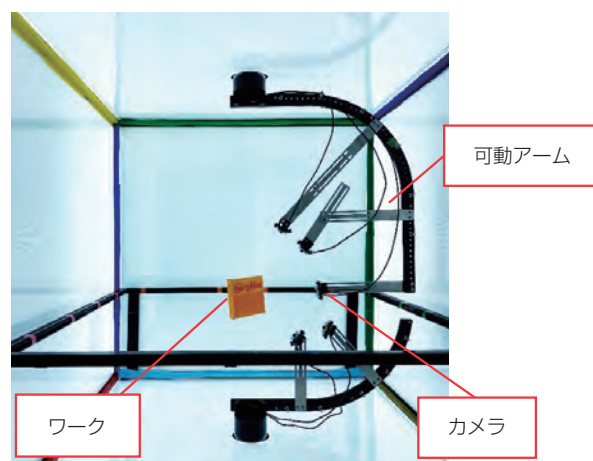


図 3. フォトグラメトリ用画像撮像装置の内部写真

大量に撮像し学習画像（バーチャル学習画像）として利用できるか検証を行うこととした。

本報告では、バーチャル学習画像生成のための 3D モデル生成手法と、カメラパラメータの設定方法について検討し、実際に、仮想環境で撮像した学習画像で判別モデルを学習させたので、これらについて報告する。

2. 課題

画像から 3D モデルを算出し仮想環境を構築する上で、仮想環境上から得られる画像と実環境から得られる画像を比較することが重要となる。

本研究では、仮想環境上で 3D モデルを配置し、仮想的なカメラにより対象ワークを撮像することで、学習画像として用いていく。精度の良い仮想環境画像を得るためには、精度良い 3D モデルを作成することが重要である。3D モデルの生成にはフォトグラメトリを用いるため、対象物以外にも特徴となる部分が写っていることが望ましい。また、人の手によって対象ワークを撮像する方法も存在するが、多量のワークを撮像することを考えると自動化が必要となる。

また、実際の判別を行う上で、バーチャル学習画像と実際の判別画像ができるだけ同じように撮像される必要がある、仮想環境上ではカメラの個体差もなく、外乱の少ない環境であるが、実際に判別する画像はカメラ単体の誤差も外乱も大きい。この差をできるだけ埋める必要があると考えた。

本報告では、対象ワークを自動的に撮像し 3D モデルを精度よく生成することと、バーチャル学習画像と現実の画像の色味を合わせることにフォーカスして検討を行う。

3. 3Dモデルの作成

前述した人手でのワークの撮像の手間を省くために、自動で対象ワークを多角的に撮像し、精度よく 3D モデルを生成する方法について述べる。

3.1 フォトグラメトリ技術

対象となる物体を 3D モデルとして取り込む方法にはフォトグラメトリ技術を採用する。フォトグラメトリを採用したのは、物体の形状以上にその表面に形成されるメッシュが判別の根拠となると考え、メッシュがきれいに再現される本技術が適合すると考えたからである。フォトグラメトリとは、複数の写真より解析統合することで 3D モデルを作成する手法である。この手法の特徴として、通常の写真から 3D モデルを生成できることが挙げられる。原理として、各画像に対して同じ個所（特徴点）を算出し、カメラから対象ワークの距離を割り出すことで、モデルを算出できる。人間の目が左右で少しずれた画像を認識して、奥行きを計算する原理である。

3.2 装置の概要

3D モデル作成のために必要な画像を自動で取得する装置を製作した。本装置では、中心に撮像対象となるワークを設置し周囲に設置したカメラにより、画像の取得を行う（図 3）。

格子状に張り巡らせた釣り糸の上に対象ワークを置くことによって、ワークの下面を撮像することも可能としている。透明の亚克力板も検討したが、亚克力板上に対象ワークを乗せて撮像すると、亚克力板にワークの影が反射して映りこみ、3D モデル生成時に悪影響を及ぼしたため棄却した。



図 4. 撮像画像（側面撮像）

照明には導光板を用いることとした。導光板は加工を施したアクリル板の端面より LED を入光させることで、光を拡散させ、面で光らせる照明である。導光板の配置を前後左右上下の 6 面とすることで、全方向からの照射が可能となり、ワークに影を作ることなく撮像が可能である（図 4）。

カメラは、上下の中心を軸とする 360 度回転可能なアーム上に複数設置することで、中心に設置されたワークに対して多角的に撮像できる。アームは、モータと接続しており外部から回転させることが可能となっている。

3.3 フォトグラメトリへの対応

3.1 項でも述べたように、フォトグラメトリでは画像の特徴を基準として 3D モデルの作成を行う。前節で採用した導光板での環境では、ワークの背面の特徴が少なくなってしまう、モデルの作成が困難である。そのため、導光板の押さえでもある装置の構造体の各辺に異なる色の色紙を貼って背景の特徴点を増加させる。

また、カメラをワークに近づけたり遠ざけたりすることができる調整板を可動アームに取り付けている。フォトグラメトリでは、3D モデル生成時に撮像された画像のテクスチャを貼り付けるため、できるだけワークの特徴を大きく映していることが望ましい。そこで図 4 のようにワーク毎にカメラを調整し、画角一杯に映すようにしている。

3.4 フォトグラメトリを用いた 3D モデルの生成

実際に前項までの装置を用いて画像の収集を行った。カメラは 3.6 度ごとに 1 枚、1 周で 100 枚撮像できる。カメラは可動アーム上に 5 台取り付け、周囲一周



図 5. フォトグラメトリで作成した 3D オブジェクト

あたり 500 枚の画像を取得することができる。本装置で、1 アイテムの画像を取得するのに要する時間は約 5 分であった。ここで、撮像した画像を用いて、フォトグラメトリの技術を利用して作成した 3D モデルが図 5 である。AI を用いた画像認識では人目で違いが判断できる画像が基準となる。作成した 3D モデルは多少表面に凸凹があったり、箱の特徴の少ない部分に本来ないはずの盛り上がりがあったりするが、この基準を満たしたパッケージの再現ができたと考える。

4. カメラパラメータの調整

物体判別 AI の学習と判別に用いる画像は、撮像画像の色味・明るさが同一であることが好ましい。しかし、撮像装置や設置環境が異なると、装置の照明や組み立て誤差、外乱の影響により撮像画像の色味・明るさを同一にすることは難しい。ましてや仮想環境と現実となるとかなり困難と予想される。環境の違いから生じる差としては、仮想環境では理想状態で撮像できるため、外乱や照明の発光のバラつきが存在しない。そのため実環境とは、存在する外乱の種類と量に差が多くなる。仮想環境と実環境で撮像される画像を近いものにするには、仮想環境上にも外乱を再現する手法もあるが、すべてを想定することが難しいとして、撮像される実画像の色味・明るさを、できるだけバーチャル画像に近づけるようなカメラパラメータ調整手法について述べる。

画像の色味を合わせる方法として、撮像された画像を調整する方法もあるが、今回はカメラの撮像パラメータである ISO・レッドゲイン・ブルーゲインを自動的に調整する。これは撮像後の画像処理以上に、撮像時のパラメータの影響が大きいと考えたためである。

パラメータ調整は、オートゲインで行うのが一般的である。しかしオートゲインは画像全体でパラメータの調整を行うため、背景の影響によって判別対象の色・明る



図 6. カラーチェッカーボード

さが変わる問題があった。そこで新たにカメラパラメータ調整手法を開発した。パラメータ調整は ISO 感度、ブルーゲイン、レッドゲインの順に行う。まず、ISO 感度を調整することによって画像の明るさを合わせる。次にブルーゲイン、レッドゲインを調整することによって画像の色味を合わせる。

4.1 色基準の作成

色合わせの基準として、カラーチェッカーボード（図 6）を用いる。まず 24 色のカラーチェッカーを撮像し、フォトグラメトリの技術を用いて 3D オブジェクトを作成する。カラーチェッカーの 3D オブジェクトは前章で述べた装置で作成する。撮像環境として 6 面共に導光板の照明で囲んでいるため、各カメラの撮像条件の差異が小さくなる。この環境下で 3D オブジェクト作成に必要な画像撮像の際は、ISO、レッドゲイン、ブルーゲインは AUTO で撮像する。

次に 3D オブジェクトから 24 色それぞれの輝度値を算出する。輝度値の算出は、3D オブジェクトから対象色部分をトリミングし、RGB それぞれの輝度値平均を算出する。この輝度値を各カメラの調整目標の色とする。

この色基準に合うように次項で述べるパラメータ調整を実施することによって、撮像環境、カメラ、ワークが異なっても同じ色味で撮像することができる。

4.2 ISO感度調整

ISO 感度の調整は、レッドゲイン、ブルーゲインを AUTO にし、ISO 感度のみを変化させてカラーチェッカーを撮像する。各 ISO 感度で撮像したカラーチェッカーの画像からグレースケールの各色の輝度値を算出する。グレースケール各色と対応する基準色の 3 次元距離 D を式 (1) にて算出する。また、3 次元距離に D の各色の平均値を $D_{平均}$ 式 (2) として採用し、基準値とする。グレースケール各色と基準色の 3 次元距離の平均 ($D_{平均}$) が最も小さくなる ISO 感度をカメラパラメータとして

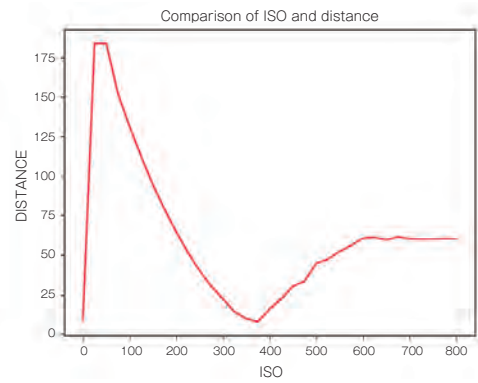


図 7. ISO 感度と $D_{平均}$

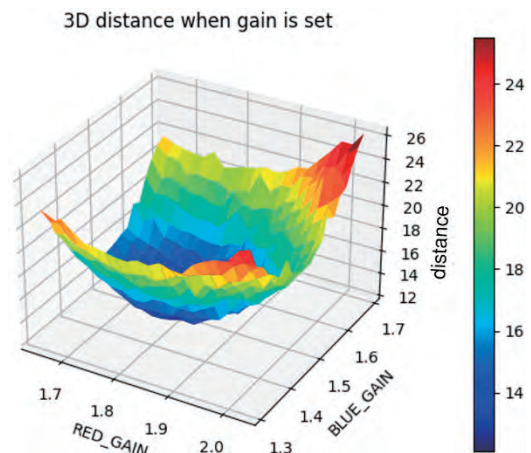


図 8. ゲインと $D_{平均}$

採用する。ISO 感度を変化させたときの基準値との ($D_{平均}$) を図 7 に示す。

$$D = \sqrt{(R_{基準} - R_{撮像})^2 + (G_{基準} - G_{撮像})^2 + (B_{基準} - B_{撮像})^2} \quad \dots (1)$$

$R_{基準}$: 基準色の R チャンネルの輝度値 (G, B 同様)

$R_{撮像}$: 撮像画像の R チャンネルの輝度値 (G, B 同様)

$$D_{平均} = \sum D / \text{距離の算出に使用した色数} \quad \dots (2)$$

4.3 ブルーゲイン調整

ブルーゲインの調整では、ISO 感度を前節で採用したパラメータに固定し、ブルーゲイン、レッドゲインのみを変化させてカラーチェッカーを撮像する。前節と同様に基準色との 3 次元距離を、カラーチェッカー上の 24 色全てに使用し算出する。ブルーゲイン、レッドゲインを変化させたときの基準色との 3 次元距離の平均を図 8 に示す。カラーバーは、偏差が大きくなるにつれ赤に近づき、偏差が小さい方が青になるよう表示してある。図 8

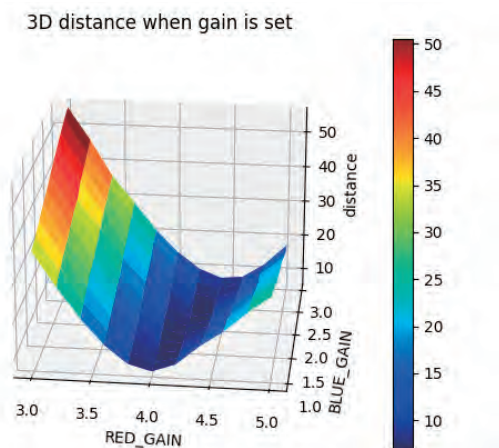


図 9. ゲインと $D_{平均}$ (Rチャンネルのみ)

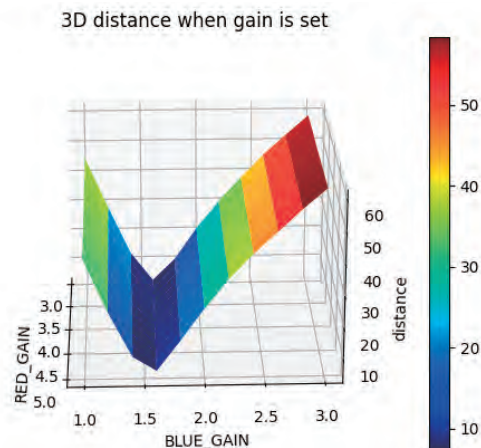


図 10. ゲインと $D_{平均}$ (Bチャンネルのみ)

を R,B チャンネルそれぞれに分解した際の距離差の結果を図 9、10 に示す。

図 8 から B チャンネルについてはブルーゲインによって基準色との距離の最小がレッドゲインに因らないことがわかる。そこで、先に距離の平均 ($D_{平均}$) が最小となるときのブルーゲインをカメラパラメータとして採用する。

4.4 レッドゲイン調整

レッドゲインの調整は、ISO 感度を 4.2 節で採用したパラメータに固定、ブルーゲインを 4.3 節で採用したパラメータに固定し、レッドゲインのみを変化させてカラーチェッカーを撮像する。各レッドゲインで撮像した画像から 24 色それぞれで R チャンネルの輝度値を算出する。この時の 24 色それぞれの輝度値と対応する基準色の R チャンネルの輝度値の差分を算出し、その総和を求める。この総和の絶対値が最小となるレッドゲインをカメラパラメータとして採用する。



図 11. コンベヤ撮像装置

クを撮像できるようにしている。

仮想環境側の装置として、3 章で用いた環境によりワークの画像を取得し 3D モデルを生成する。Unity^{*1}内 (PC 内の仮想環境) にコンベヤ撮像装置と同じ配置にしたコンベヤおよびカメラを用意しバーチャル学習画像を生成した。バーチャル上で作成した画像の一部を図 12 に、実環境で撮像した画像の一部を図 13 に示す。

生成したバーチャル学習画像で物体認識モデルを作成し、検証用画像を撮像し、前述の判別モデルに判別させた。

5. 判別結果

前章までに述べた手法を利用して、3D モデルを作成し、仮想環境上での学習画像の撮像することで、特定物体認識の判別を行う。

5.1 検証方法

仮想環境と実環境での融和性を確認するために、仮想環境で作成した学習モデルを用いて、実環境上の画像で認識が可能か確認していく。

始めに、実環境の装置として、図 11 に示すコンベヤ撮像装置を作成した。コンベヤ上で導光板に囲まれた環境内にカメラを複数設置しコンベヤにて搬送されるワー

5.2 判別結果・考察

今回の検証に用いたワークは 3D 形状を再現しやすい、箱形状のものに限定して 20 種類とした。その結果を表 1 に示す。正解率は 88% となった。

良好な結果が得られたのは、前項までの工夫とモデル生成時に使用した照明と、判別時に使用した照明が同種のものを利用しており、ワーク表面に発生する照明の反射が近いものになったことがあげられる。一方で、正解率が約 88% に留まったのは、Unity 上でコンベヤ撮像装置を再現したが、ワーク表面の反射性などに実物と差異があることが考えられる。また、ワーク 1 種類に対して

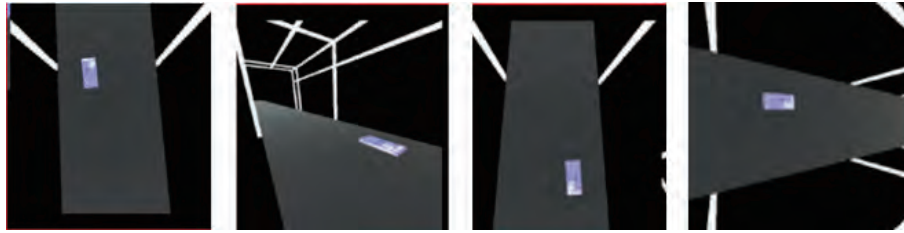


図 12. バーチャル学習画像（抜粋）



図 13. 判別画像（抜粋）

表1. 判別結果

予測数	正解数	不正解数	正解率 (%)
184	161	23	88.67

3D モデルは一つしか作成していなかったため、3D モデル上の凸凹のようなノイズをワークの特徴として学習してしまったことが原因であると考える。

技術用語解説

*1 Unity

Unity Technologies が開発・販売している、IDE を内蔵するゲームエンジンである。

他社商標

* 「Google」は「Google Inc.」の登録商標であり、「Google レンズ」は「Google Inc.」の画像検索サービスです。

6. おわりに

本報告では、特定物体認識を行うための機械学習に必要な、学習画像の収集方法に関して、3D モデルを作成し多数の学習画像を作成する方法について述べた。実際に仮想環境上のバーチャル画像から学習モデルを作成し、実環境で判別を行った結果 88% の正解率となった。この値を、実際の製品上での判別に利用するには精度を上げていく必要がある。

今後は、正解率向上やワーク種を増やす活動を続けつつも、疑似学習画像による正解率には、限界があることもわかってきたので、仕分け装置だけに限らず本技術の適応アプリケーションを提案しつつ、確立させていきたい。本技術の根幹は仮想環境で生成したデータで学習を行い実環境に展開できることであるので、学習データの取得が難しいケースにも対応できるように発展させていきたい。

参考文献

- 1) 梅村良佑, 松村翔太, 物体判別装置における撮像機の検討
TSUBAKI TECHNICAL REVIEW Vol.31 No.1 2021 27-30

リニソート® トレー上物体検出装置の開発

Development of Linisort tray object detection

平井 達也^{※1}
Tatsuya HIRAI河野 浩二^{※1}
Kouji KOUNO

マテハン製品であるリニソートのトレイ上物体検出を画像処理で行い、かつ安価な構成を提案する。カメラで物体を検出可能なセンサとみなし本用途以外での使用も視野に入れ、工場などの環境下でも使用に耐えるよう画像処理以外の部分にも配慮した構成にしている。センサからの撮像指示入力から撮像までのタイムラグを大幅に削り、熱による電子機器への影響を極力減らす工夫をしている。最大120m/minの移動物体を撮像し、100ミリ秒以内に撮像指示～画像処理～結果送信まで行う。

We propose an inexpensive configuration that uses image processing to detect objects on the tray of Linisort, a material handling product. By considering the camera as a sensor capable of detecting objects on a plane, the camera is also designed to be used for applications other than this purpose and is configured to withstand use in environments other than image processing, such as in factories. The time lag between the input of imaging instructions from the sensor and the image capture is drastically reduced to minimize the heat impact on electronic devices. It is capable of capturing images of objects moving at a maximum speed of 120 m/min and performs image capture instructions, image processing, and results transmission within 100 msec.

エコロジー訴求点と
顧客メリット

- ①光電センサ検出範囲外（薄いワークなど）に対する優位性があり、検出漏れが減る
- ②季節や温湿度の影響によるセンサの調整にかかる人件費・交通費の削減
- ③民生品を産業用途で使用できれば大幅なコストダウンになる

1. はじめに

リニソートはトレイ群が周回して仕分けを行う装置である。仕分けを高速かつ効率的に行う装置であり、人手不足が課題の物流業界において省人化・効率化に貢献する装置である。仕分け対象品を投入する前後など装置の複数個所に、トレイ上の物品有無を確認するための透過型あるいは反射型光電センサが設置されている。光電センサは光の特性を利用して多種多様な物体の有無などを検出することのできるセンサで、比較的安価に利用することができる。一方で、微小なものや厚みの薄い物体の検出は困難であり、リニソートでも微小物体や薄い物体の検出のために微妙な調整が必要であった。そこで画像センサによってトレイ上の物体を検出する技術開発を進めることとなった。画像センサ（カメラ）は多数の受光素子が格子状に配列しており、光を検出し2次元の画像を処理・解析することができる。画像センサを用いれば光電センサでは対応困難な対象物体にも対応可能である。

画像センサは物体検出に有効であるが工業用の画像処理システムは高額で光電センサの代替品と認識されない。そこで光電センサよりも高価であるが民生品のWEBカメラや小型コンピュータを用いて画像処理を行い微小物体や薄い物体の検出を行った。民生品を産業用途で使用できれば大幅なコストダウンが期待できる。

2. 概要

2.1 要求仕様

表1. 要求仕様

トレイ最大移動速度	120m/min
トレイピッチ	600～800mm
許容画像処理時間	100ms以内
対象ワーク	ボールペンサイズ（150×10×10mm）～ 目薬（50×50×20mm）～ 紙おむつ（300×300×170mm）

2.2 過去の課題

現在のリニソートは光電センサを用いた検出が一般的

※1 研究開発センター 先端技術部

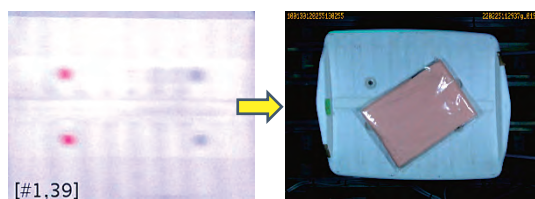


図 1. 不明瞭な撮像の改善

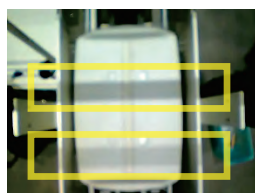


図 2. 天井照明の影響

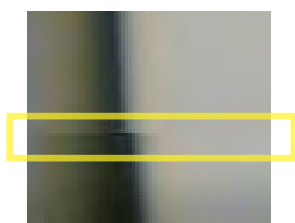


図 3. カメラ画像が不連続（拡大画像）

であり、微小なものや厚みの薄い物体の検出は困難である。微小物体・薄い物体への対策として画像センサを用いた物体検出を検討した。最初の画像センサを用いた検出は市販の画像センサシステム一式を用いたが経年変化によるトレー汚れや外光による影や色味の影響を受け、検出精度が落ちた。原理的には比較元画像として1種類のトレーを登録し物体が乗ったトレー画像との差を物体の有無としたため、トレー毎に異なるトレー汚れに対処できなかった。次に当社独自の安価な画像処理システムを構築したが、撮像が不明瞭な事象（図1参照）や、天井照明の影響で影が映る事象（図2参照）が発生し、物体検出に悪影響を与えた。またセンサからの撮像指示が入ってもトレーが数枚通過して撮像するなどトレー撮像の大幅な遅延が生じた。画像センサで撮像した画像の中には不連続な部分が見られた（図3参照）。要求仕様の許容画像処理時間 100ms の達成も困難であった。

2.3 課題の解決策と開発方針

まず光電センサで微小物体や薄い物体が検出困難な理由として光電センサでは一カ所のみを検出するためセンサの検出範囲外の物体は検出できない。トレー上のエリアを面状に画像として検出可能な画像センサを用いることで課題の解決を図る。また画像センサは格子状に配列された受光素子センサの集合とみなし、センサからのデータ量

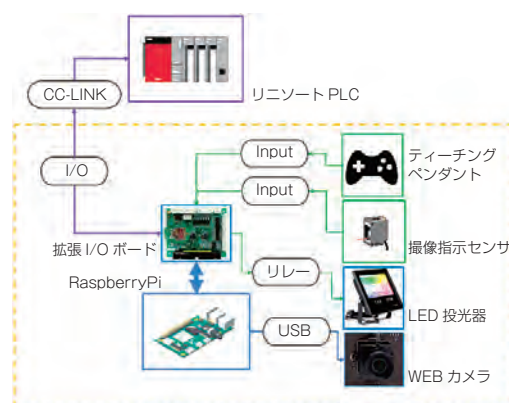


図 4. 基本的な構成

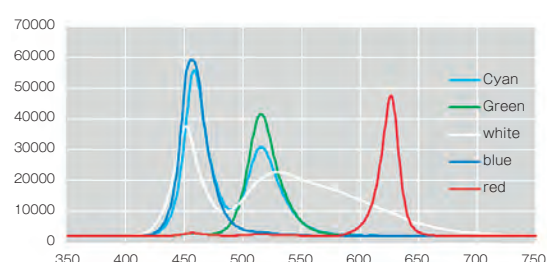


図 5. LED 波長分布



図 6. リニソートと照明+暗幕

が多く、複雑な計算式を用いて多面的・複合的な判断が可能である。次に各トレー汚れの違いに対応できなかった理由は画像比較処理用の元画像データを1種類の空トレーとし複数の撮像画像データと比較するためである。そのため画像比較処理をトレー毎に個別に実施することでトレー汚れに対処可能である。不明瞭な撮像（図1参照）は画像センサの露光時間内にトレーが最大 120m/min の速度で移動し検出物体が進行方向に歪んで撮像されることが原因である。微小物体やトレーと同色系物体を検出困難にする。その対策として高速撮像、具体的には画像センサの露光時間を短くすることが有効である。半面、露光時間を短くすると撮像に必要な光量が不足し、暗い画像となるため照明投光を行う。次に天井照明・西日など外光

の影響（図2参照）はトレー上に生じた装置の影を検出物体と誤認識するため、影をトレー上に写さないように照明投光と部分的に上方からの遮光を行い外光の影響を押さえる。過去に実施した当社独自の安価な画像処理システムとして Raspberry Pi^{*1} というシングルボードコンピュータを使用した。今回も同様に Raspberry Pi にて検討する。過去の検討では撮像遅れの課題があり、原因として3つ類推される。1つ目は撮像指示センサ入力～撮像までの時間のバラツキが大きく、トレー撮像位置のズレが生じる課題である。画像比較処理時に比較元画像と撮像画像のトレー位置が大きくずれる場合は比較処理が煩雑になり、処理時間も長くなる。対策として撮像指示センサ入力した時に即時撮像指示する割り込み処理を行い対処する。撮像遅れの2つ目の原因は熱による Raspberry Pi 自身の低速運転モードへの移行である。常時画像処理することで Raspberry Pi 自身が発熱し、保護のため低速運転モードに移行するが、画像処理自体も低速になり処理が間に合わなくなる。対策として低速運転モード禁止措置と熱による破損防止の冷却を行い常時通常運転モードによる画像処理を実行する。3つ目の原因は記憶媒体の常時読み書きによって Raspberry Pi 内のプログラム処理時間の大部分が読み書きに割かれて画像処理しきれない状態になっていることが原因と考える。対策として Raspberry Pi の記憶媒体を高速通信可能なものに置換することで読み書きの通信負荷を軽減する。また撮像データの不連続性（図3参照）は Raspberry Pi や画像センサの稼働負荷が高くなったため画像データの欠損が起こったと考えられる。先述の Raspberry Pi や画像センサへの稼働負荷を軽減することで改善されると考える。

そこで開発方針は以下の6つとした。

- ① 画像センサを使った物体検出
- ② 画像比較処理による物体検出
- ③ 遮光・照明投光による外光の影響排除
- ④ 高速撮像・短時間露光による明瞭な撮像
- ⑤ 撮像指示センサ入力後の即時撮像指示
- ⑥ Raspberry Pi の負荷軽減（CPU・通信・熱対策）

3. 構成要素

3.1 基本的な構成（図4参照）

安価かつ仕様を満たす性能の部品で構成する。画像処理システムは画像処理装置・カメラ・照明で構成される。画像処理プログラムは安価なコンピュータ Raspberry Pi で処理し、カメラは産業用カメラではなく安価な民生品 WEB カメラを使用する。WEB カメラは画像の高速転

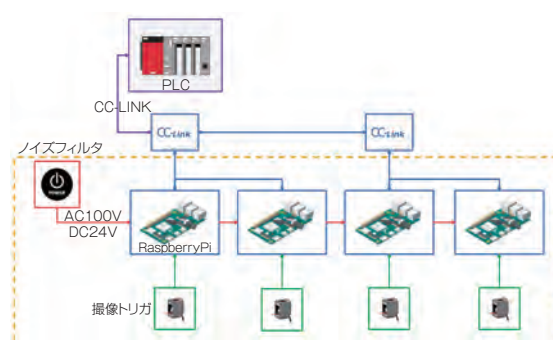


図7. リニソート PLC と CC-Link でデータ通信

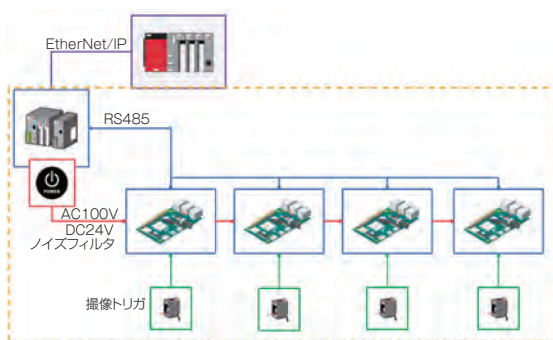


図8. リニソート PLC と通信用 PLC

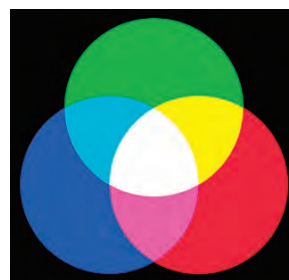


図9. Red・Green・Blue：光の三原色

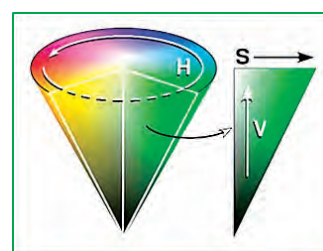


図10. H：色相・S：彩度・V：明度

送に適した USB3.0 で Raspberry Pi と接続し、高速撮像に適した 50fps で撮像できるものを選んだ。照明もまた産業用の均質化された光を投光できる照明でなく、安価な LED 投光器を使用する。LED 素子は RGB W（赤緑青白）（図5参照）の光量調整が電流制御可能なものを選定した。照明の点灯は Raspberry Pi 拡張 I/O ボードを介してリレー接点に行う。また装置を暗幕で覆い外光の影響を避けている（図6参照）。リニソート

PLC*2と画像処理システムは物体の有無信号・測定準備完了信号・撮像指示信号・周回基準信号（1周に1パルス）・測定モード切替信号の送受信を行う。図4.黄色枠内が今回の開発範囲である。

3.2 リニソート装置との接続

【基本接続例】図4・図7参照

リニソート PLC と複数の Raspberry Pi が個別に接続する方式。具体的には CC-Link*3 などのネットワークを通じてリニソート PLC と Raspberry Pi を接続する。CC-Link 経由で I/O*4 通信する。少数のトレイ上物体検出装置を設置する場合に向く。ティーチングペンダントはトレイ上物体検出装置1台につき1カ所接続する。

【設置台数増加時】図8参照

リニソート PLC と複数の Raspberry Pi をまとめて接続する方式。具体的にはリニソート PLC との通信専用 PLC を用意し、通信専用 PLC と Raspberry Pi が RS485*5 で通信する。トレイ上物体検出装置の設置台数が2台以上併設されている場合など各装置への設定を通信専用 PLC で管理可能なため複数台同時設定が可能となる。リニソート PLC との接続 I/O の節約・省配線にも貢献する。ティーチングペンダントはトレイ上物体検出装置複数台に対し1カ所接続する。なお、ティーチングペンダントは通信専用 PLC で代用可能である。

4. 開発方針と開発方法

4.1 画像比較処理について

“2.3 解決策と開発方針”で述べた6つの開発方針の内、①②の具体的な内容を説明する。光電センサで検出困難な微小物体・薄い物体は画像比較処理を用いてトレイを面状に検出することで検出が容易になる。画像センサはカラーのWEBカメラを使用し、トレイの比較元画像とトレイ撮像画像を比較して物体検出を行う。カラーのWEBカメラから取得できるRGB三原色の画像情報(図9参照)を人間の色覚に近いHSV画像*6に変換(図10参照)する。HSV画像情報はRGB画像情報より照明の変化や色味の変化に強い。ためHSV画像情報を用いて画像データ比較を行う。処理時はトレイ一枚ごとに比較元画像をあらかじめ取得し、トレイ番号に応じた比較元画像データと撮像画像データの比較を行う。実際のデータ比較は画像比較処理プログラムの高速化のため、最初にエリア分割ブロック処理と間引き処理を行う。まずエリア分割ブロック処理は12×8画素のブロックに区切ってブロックの代表値を比

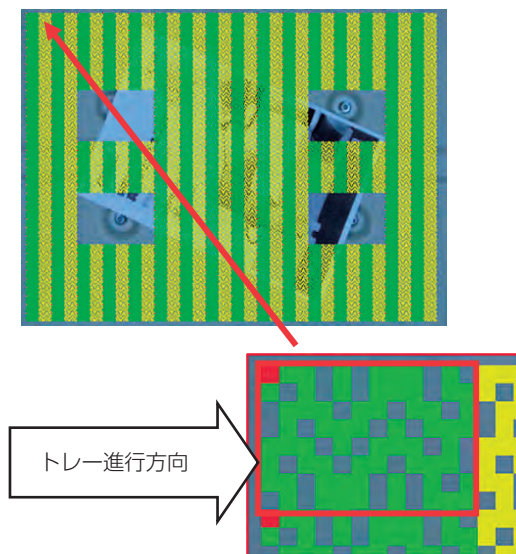


図 11. エリア分割と画像データ中抜き

$$R_{NCC} = \frac{\sum_{j=0}^{N-1} \sum_{i=0}^{M-1} I(i, j) T(i, j)}{\sqrt{\sum_{j=0}^{N-1} \sum_{i=0}^{M-1} I(i, j)^2 \times \sum_{j=0}^{N-1} \sum_{i=0}^{M-1} T(i, j)^2}} \quad \dots (1)$$

較する。具体的には図10の色相・彩度・明度のブロックごとの代表値を正規化相互相関*7と言う手法(式(1)参照)で比較する。各画素を比較するよりもブロックの代表値を比較することで比較データ数が1/96(=12×8)に減り、かつトレイ位置ズレやトレイ回転ズレに対しても鈍らせた画像データは有利である。結果画像比較処理の高速化と計算量の節約になる。なお外乱を受けやすいボルト穴位置などは除外する。次に12×8画素の画像データを8×8に間引く。隣り合ったブロックで画像データの集中的な欠損が起きないようにジグザグに隣り合ったブロックと連続する間引き方法を行う(図11参照)。ブロック内の有効な画素数は8×8=64で2の6乗であり、画像データの演算処理を行うに当たってシフト演算を適用しやすい数字にしている。またブロックごとの計算データ数が96画素から64画素と2/3に減らせる。トレイ進行方向への配慮として、トレイ進行方向に画素数が多い12×8画素のブロックを規定し、正規化相互相関も進行方向に柔軟性を持たせるように比較ブロック数を調整している。

4.2 照明環境について

“2.3 解決策と開発方針”で述べた6つの開発方針の内、③④を実施するには期待する遮光・照明位置・光量などを満足する照明投光環境を整える必要がある。照明位置・カメラ位置・トレイ位置の相対関係を図12に示す。カメ



図 12. カメラ・トレー・照明の位置関係

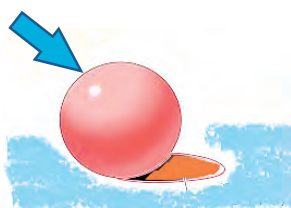


図 13. 斜め上方からの投光により物体に影が生じる

ラをトレー上部に据え、照明を斜め上方から投光する。斜めから投光することで厚さのあるものは影を作り、後述する画像比較の際に影の有無で物体検出が可能になる。透明な物体やトレーと同色な物体も影を生じさせることで物体検出が可能になる（図 13 参照）。また、投光することで露光時間を短くすることが可能になり、より鮮明な画像の取得ができる。図 6 に示すように外光の影響を減らすため、カメラ・照明の上部・斜め上部を暗幕で覆う。

4.3 画像処理時間と撮像遅れについて

“2.3 解決策と開発方針”で述べた6つの開発方針の内、⑤⑥の具体的な内容を説明する。リニソートの仕様としてトレーのサイズと移動速度より算出して1秒あたりに2～3枚のトレーが画像検出部を通過する。前記、“2.2 過去の課題”で言及したトレー撮像の大幅遅延は撮像指示の入力があったからトレーが数枚通過後に撮像するという問題で秒換算すると1～3秒程度の遅れが発生していることになる。要求仕様による画像処理に割ける時間は100msのため大きな課題である。遅延が発生する理由の一つは熱によるRaspberry Piの低速運転モードへの移行である。そこでCPU動作周波数を通常運転モードに固定し、高熱によって破損しないよう空冷を行う。また常時実行されている画像処理プログラム内のそれぞれの処理時間を比較検証することで遅延に対して効果的な対処方法を類推する。トレーが画像検出部を通過する度に画像処理プログラムが実行されるが、次のトレーが通過するまでの時間内にプログラムが終わらない場合は終

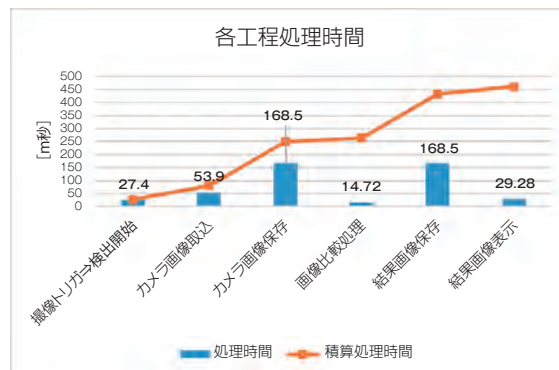


図 14. 改善前の画像比較処理にかかる時間

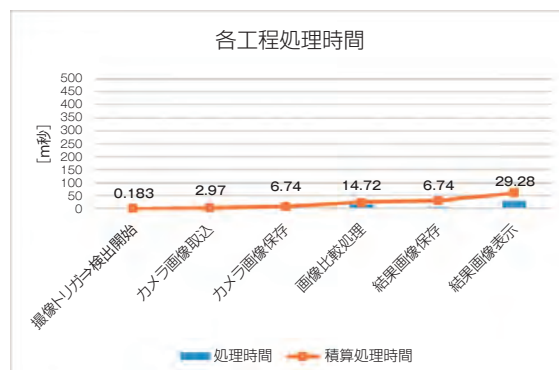


図 15. 改善後の画像比較処理にかかる時間

わっていない処理が積算して大幅な遅延につながる。画像比較処理は余裕をもって時間内に処理が完了することが望ましい。実際には画像比較処理は以下の処理プログラムが繰り返し実行される。

- ・撮像指示センサ入力～カメラ画像検出開始
- ・カメラ画像取込
- ・カメラ画像保存（デバッグ用）
- ・比較元トレー画像データの読込と撮像データとの比較
- ・結果をリニソート PLC へ出力・結果画像保存
- ・結果画像表示

図 14、15 中の処理時間は最大処理時間を表している。処理に一番時間のかかっているのは“カメラ画像保存”と“結果画像保存”である（図 14 参照）。画像ファイルの保存に時間がかかっていることが判明したので Raspberry Pi の記憶媒体を高速なものに変更することで改善する。具体的には SD^{*8} カードを SSD^{*9} に変更する。結果画像データの読込・書込み速度が 168.5ms から 6.74ms と約 25 倍程度改善される（図 15 参照）。次に“カメラ画像取込み”時間について考える。選定した WEB カメラは撮像速度と通信速度の両方を高速で処理できるものを選んでいく。しかし画像処理中、カメラから Raspberry Pi への画像データ転送に 53.9ms かかっている（図 14 参照）。50fps の WEB カメラは 20ms ごとに撮像

できるが、今回の画像転送には倍以上の時間がかかっている。これは WEB カメラの仕様で高速に動画を撮像するためにあらかじめ何枚も画像を撮像済みであることが原因と類推する。あらかじめ保存された画像を読み出した後でないと撮像指示後の画像が転送されないため、あらかじめ撮像画像を保存しないカメラ側の設定・Raspberry Pi 側の設定・画像処理の設定を行う。結果カメラから Raspberry Pi への画像データ転送時間は 53.9ms から 2.97ms に改善される。因みに通常のデフォルト設定ではカメラから Raspberry Pi へのデータ転送時間は約 120ms、最小値を設定した場合には 53.9ms である。動画の様に連続撮像する場合には問題ない仕様で 50fps を実現可能であるが今回のように撮像指示信号を入力して 1 枚だけ撮像するという用法では各種設定を最適にする必要がある。画像データ保存とカメラ画像取込みの対策で許容画像処理時間 100ms 以内は達成できる。最後に“撮像指示センサ入力～カメラ画像検出開始”の処理時間について考える。時間短縮への影響度はそれほど高くないが、撮像時のトレー位置がずれることへの対策として必要である。“撮像指示センサ入力～カメラ画像検出開始”はトレー通過を検出するセンサからの入力信号によって Raspberry Pi から WEB カメラへ撮像指示をプログラム上で行うまでの時間である。具体的に Raspberry Pi は画像処理以外にもさまざまな処理プログラムを実行しており、センサから入力があっても即時検出・即時対応できず、実行処理中のプログラムが処理できるタイミングまで待機することになる。その処理時間は 0 ～ 27.4ms までバラツキがある。距離換算して最大 0 ～ 54.8mm 撮像位置のバラツキがある。そこで実行処理プログラムの途中で撮像指示センサ入力があれば即時対応できるように割り込み処理を行う。撮像指示センサ入力があった場合に最優先で対応することで“撮像指示センサ入力～カメラ画像検出開始”時間は 0 ～ 0.12ms に改善できた。同様にトレーの位置ずれも最大 0.24mm に改善できた。“撮像指示センサ入力～カメラ画像検出開始”・“カメラ画像取込”・“カメラ画像保存”・“結果画像保存”の処理時間を短縮するため、記憶媒体を高速化・カメラ画像転送パラ



図 16. 簡易的なリニソートイメージ

メータの最適化・撮像指示センサ入力時の即時対応によって画像処理プログラムの処理時間は 60ms で要求仕様の許容画像処理時間 100ms 内に収めることができる。

5. 物体検出実験結果

5.1 画像による物体検出結果

リニソート試験機において実験を行った (図 16 参照)。通常のリニソートは一種類のトレーで統一されているが、試験のためさまざまなサイズ、形状のトレーを一つのリニソートに混在させて実験を行った。図 17 ～ 図 20 に画像処理によるリニソートトレーの物体検出結果の一例を示す。

図 17 はトレーと同色系の白いシャツを検出している。“4.2 照明環境について”の記述にある画像処理が苦手とする透明な物体やトレーと同色系の物体は照明の配置によって影を生じさせることで目論見通りに影を検出している。図 18 は名札入れを検出している。照明によって生じた影と照明の反射光を検出しており、照明位置や照明の設置が有効であることを示す。また、クリップ部分を検出しているのはクリップサイズの微小物体の検出も可能であることを示す。“2.3 課題の解決策と開発方針”で述べた光電センサでの検出は困難と思われる微小物体・薄い物体の検出が画像比較処理で可能なことを示す。図 19 は歯ブラシを検出している。微小物体・薄い物体の一分類の細い物体・薄い物体の検出は“2.3 課題の解決策と開発方針”で述べた光電センサに対する画像比較処理の有効性を示す。図 20 はトレーの汚れを模した模様付きのトレーを用いて薄い赤色のフィルムを検出している。トレーの模様を検出せずに赤いフィルムのみを検出している。“2.3 課題の解決策と開発方針”で述べた個別のトレー汚れ対策としてすべてのトレー比較元データを保持して画像比較処理を行っていることが効いている。また図 17 ～ 20 に見られるようにリニソートのトレーが最大 120m/min で動いても問題なく明瞭に撮像できている。高速撮像と短時間露光の効果がでている。また、“2.2 過去の課題”で言及した撮像した画像に発生していた不連続な部分 (図 3 参照) も発生が見られないので、高温対策や各種高速化処理によるプログラムの渋滞防止が効いたと思われる。

結果、光電センサなどで検出困難な薄い物体・微小物体・細い物体が画像処理を使うことで検出可能である。一方画像で検出困難と思われるトレーと同色系物体や透明系物体は照明によって影を作ることや照明光を反射させることで検出可能である。またトレー汚れの個別の違いにも対応した画像比較処理ができた。

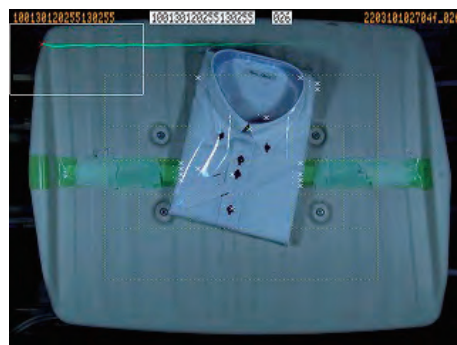
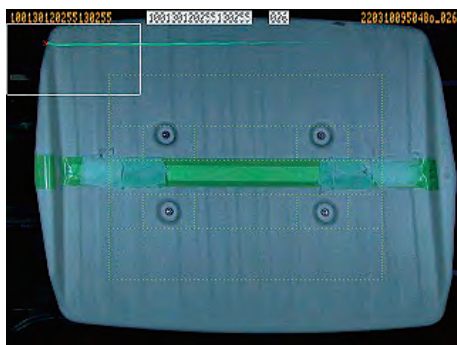


図 17. 白いシャツの検出（左：比較元画像）、（右：検出画像）影や鏡面反射部分検出

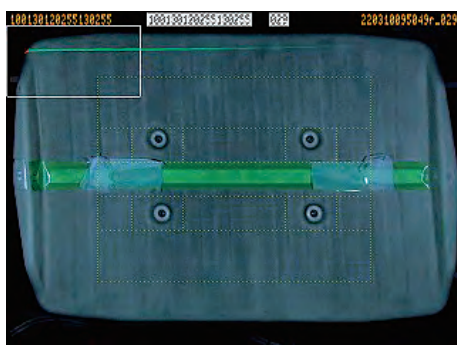


図 18. 名札入れの検出（左：比較元画像）、（右：検出画像）小さなクリップ部と鏡面反射部検出

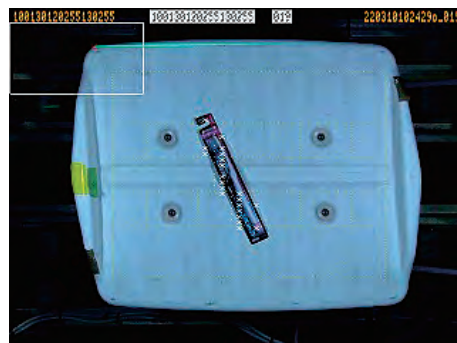
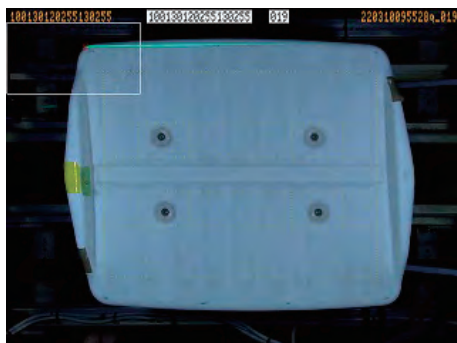


図 19. 歯ブラシの検出（左：比較元画像）、（右：検出画像）

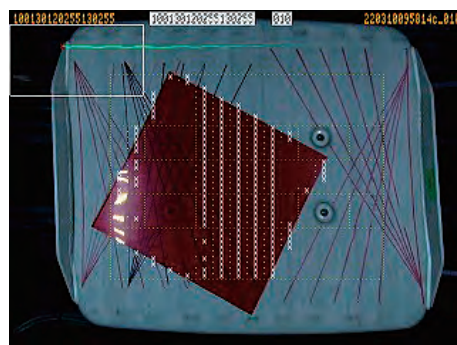
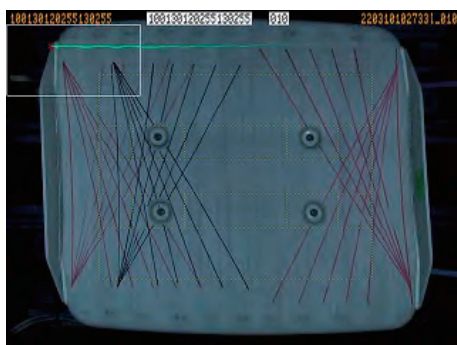


図 20. トレー汚れを無視し検出可能（左：比較元画像）、（右：検出画像）

5.2 画像処理時間

比較元トレイ画像とトレイ撮像画像の画像比較処理の最大処理時間：60ms である。画像比較処理をリニソートのトレイ上物体検出で一日連続で行い、撮像漏れや撮像遅れは確認できなかった。より長期間の確認は必要であるが、“2.3 課題の解決策と開発方針”で言及した撮像漏れや撮像遅れの原因と思われる熱による低速運転モード移行やプログラムの渋滞が発生していないと思われる。

6. 今後

本開発の基本構成である画像処理装置・照明・カメラについて要求された仕様を満たすことができた。今後は周辺機器について検討を行う必要がある。リニソートへの設置方法と併せて小型化・簡素化・省スペース化も検討する。画像処理プログラムを実行している Raspberry Pi、投光器・WEB カメラなどは工業用途での使用を想定していない民生品のため信頼性・耐久性や使用可能湿度など周辺環境について懸念事項がある。しかし安価であることを前提として予備部品を含めた構成を考え、故障の早期検出方法も必要な機能として今後検討する。

技術用語解説

* 1 Raspberry Pi

ARM プロセッサを搭載したシングルボードコンピュータ。教育で利用されることを想定して制作された。安価に入手できるシングルボードコンピュータとして趣味や業務（試作品の開発）用としても用いられるようになった。「Raspberry Pi」はラスベリパイ財団の登録商標です。

* 2 PLC

プログラマブルロジックコントローラは、リレー回路の代替装置として開発された制御装置である。プログラマブルコントローラとも呼ばれ、一般的にシーケンサ（三菱電機の商品名であるが登録商標ではない）とも呼ばれる。

* 3 CC-Link

多くのメーカーのデバイスを交信できるようにするオープンな産業用ネットワークである。多くの場合、製造業におけるセルまたはプロセス制御用途の機械に使われる。「CC-Link」は三菱電機株式会社の登録商標です。

* 4 I/O

入出力の略（Input/Output）。転じて各種の I/O ポートのこと。

* 5 RS485

2 線式、半二重、マルチポイントシリアル接続を特徴とする、OSI 参照モデルでいうところの物理層の電氣的仕様である。

* 6 HSV

HSV 色空間（英：HSV model）は色相（Hue）、彩度（Saturation・Chroma）、明度（Value・Brightness）の三つの成分からなる色空間。

* 7 正規化相互相関

パターンマッチングの一手法。

* 8 SD

フラッシュメモリーに属するメモリーカードである。SD カードとも呼ばれる。デジタルカメラ、携帯電話などの携帯機器やテレビなどの家電機器まで幅広く利用されている。

* 9 SSD

半導体メモリをディスクドライブのように扱える補助記憶装置の一種である。シリコンドライブ、半導体ドライブ、メモリドライブ、擬似ディスクドライブなどとも呼ばれる。