

MATLAB EXPO 2016

MATLABによる作業異常検知システムの開発

2016年10月19日(水)

(株)日立製作所 CTI 生産イノベーションセンタ
生産システム研究部生産S2ユニット研究員
兼 インダストリソリューションデザイン室員
今沢 慶

MATLAB EXPO 2016

MATLABによる作業異常検知システムの開発

- 1. 背景
- 2. 作業異常検知システムの概要
- 3. 開発の流れ
- 4. 応用例とまとめ

歩留まり予測

歩留まり向上が目的
センサデータと
歩留まりデータを取得
品質モデリングを実行
次工程や次着工品を制御

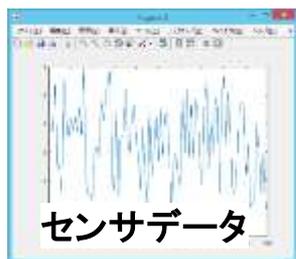
信頼性向上が目的
検査データと
信頼性データを取得
品質モデリングを実行
出荷判定や次着工品を制御

信頼性予測

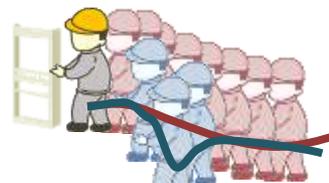
製造工程

検査工程

フィールド



ドカ不良防止や
稼働率向上が目的
センサデータを取得
例えば、温度や電力など
外れ値検出/変化点検出を実行
異常検知結果をメールで通知



動線データ

作業不良防止や
スキル向上が目的
カメラ画像データを取得
作業員検出/外れ値検出を実行
異常検知結果をメールで通知

作業異常検知

装置予兆検知

歩留まり予測

歩留まり向上が目的
センサデータと
歩留まりデータを取得
品質モデリングを実行
次工程や次着工品を制御

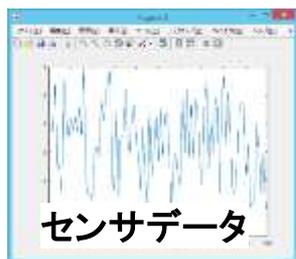
信頼性向上が目的
検査データと
信頼性データを取得
品質モデリングを実行
出荷判定や次着工品を制御

信頼性予測

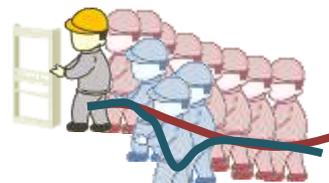
製造工程

検査工程

フィールド



今回はこの事例にフォーカスします



動線データ

作業不良防止や
スキル向上が目的
カメラ画像データを取得
作業員検出/外れ値検出を実行
異常検知結果をメールで通知

装置予兆検知

ドカ不良防止や
稼働率向上が目的
センサデータを取得
例えば、温度や電力など
外れ値検出/変化点検出を実行
異常検知結果をメールで通知

作業異常検知

MATLAB EXPO 2016

MATLABによる作業異常検知システムの開発

1. 背景

➤ 2. 作業異常検知システムの概要

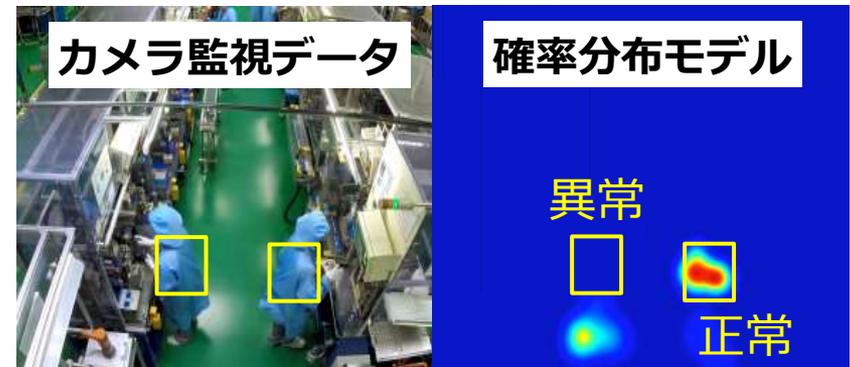
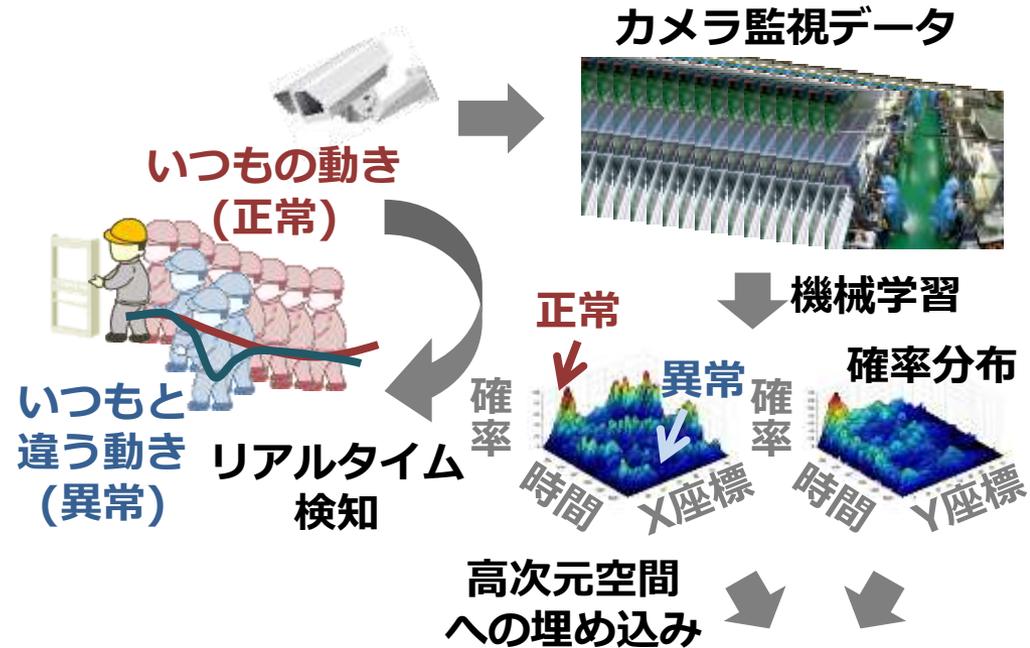
3. 開発の流れ

4. 応用例とまとめ

グローバル製造拠点における品質バラツキの解消、作業起因の場外不良の低減、品質トレーサビリティ確保によるリコール対応迅速化に貢献します。

原理説明

- カメラ監視データから作業者の動作や動線を抽出し、構造データ化します。（これにより、データ圧縮も容易になります。）
- 構造化したデータを活用した機械学習により、作業者の正常な動線や動作の確率分布モデルを生成します。
- 生成した確率分布モデルをもとに、いつもと違う動きを作業異常としてリアルタイムに検知します。（検知速度：1秒以下）

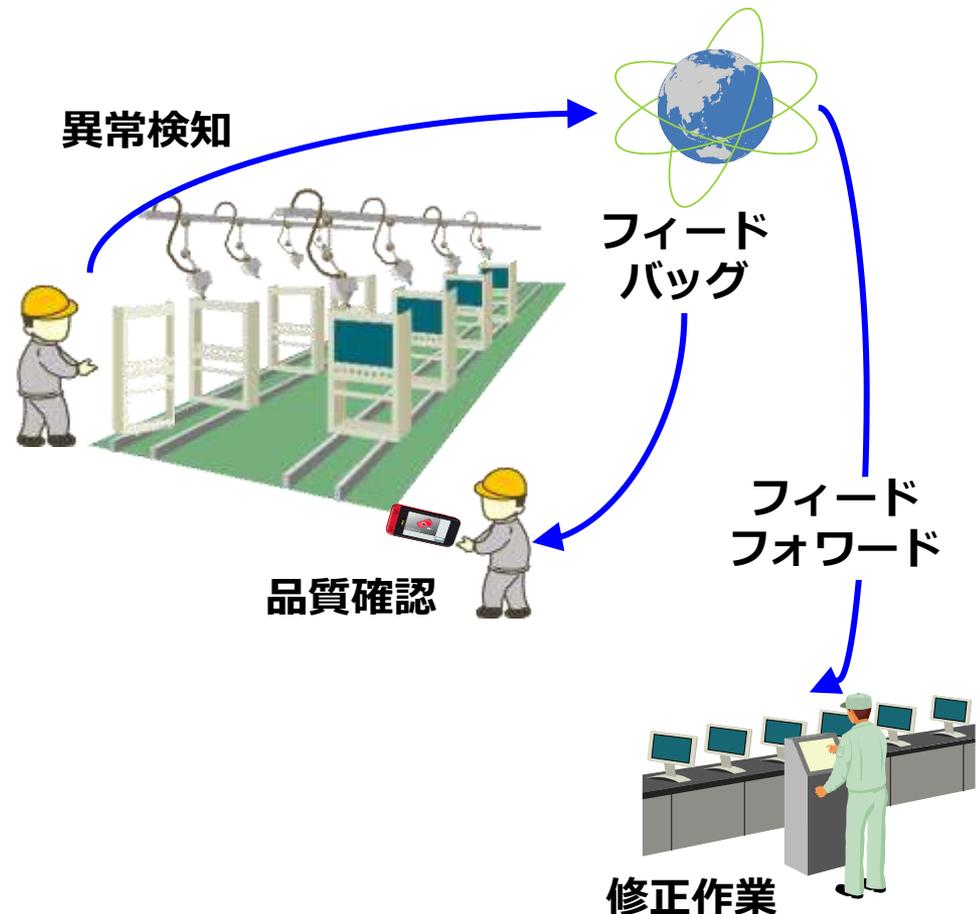


確率分布と作業者の位置を比較し、作業異常を検出

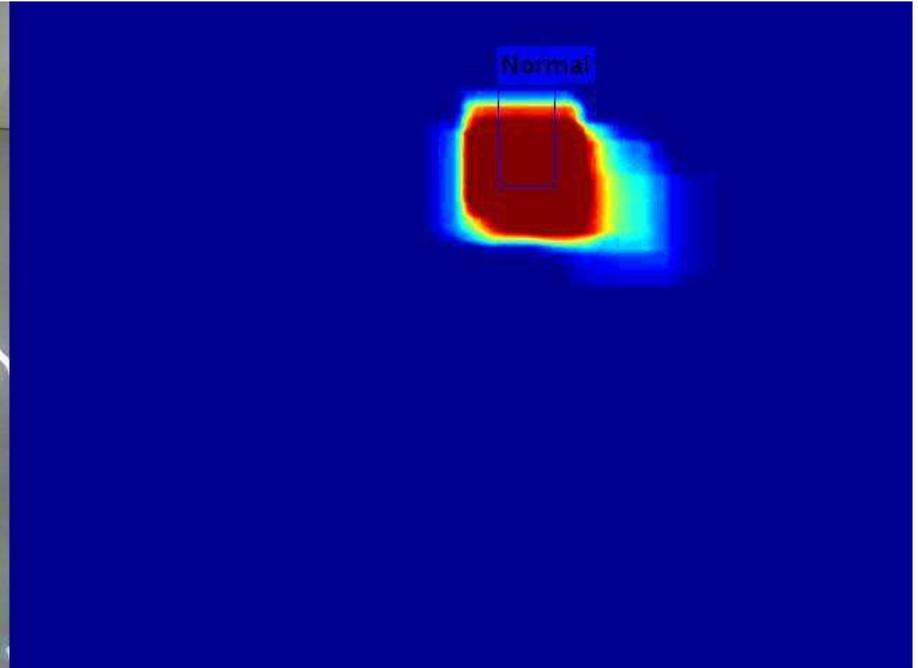
グローバル製造拠点における品質バラツキの解消、作業起因の場外不良の低減、品質トレーサビリティ確保によるリコール対応迅速化に貢献します。

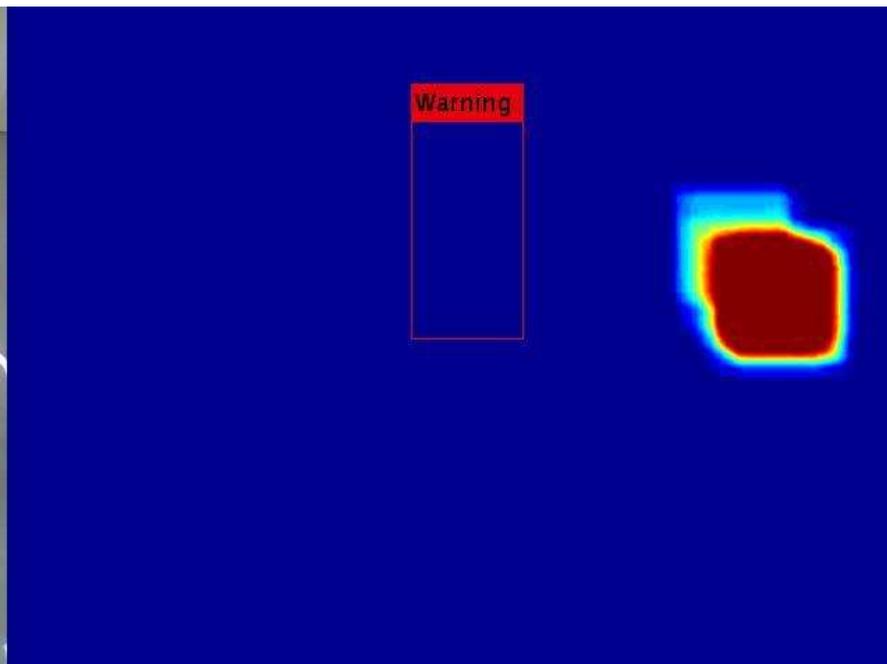
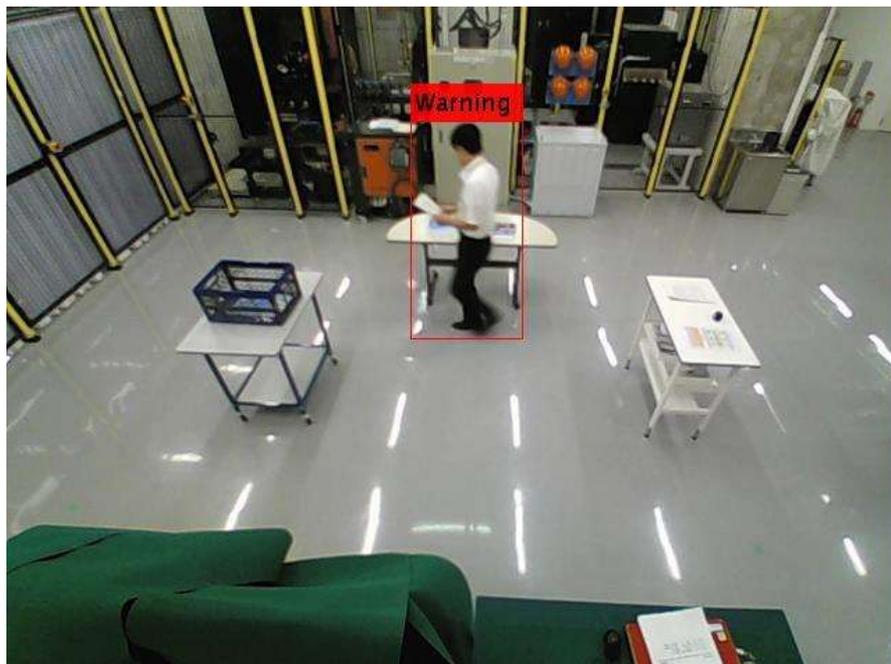
適用例

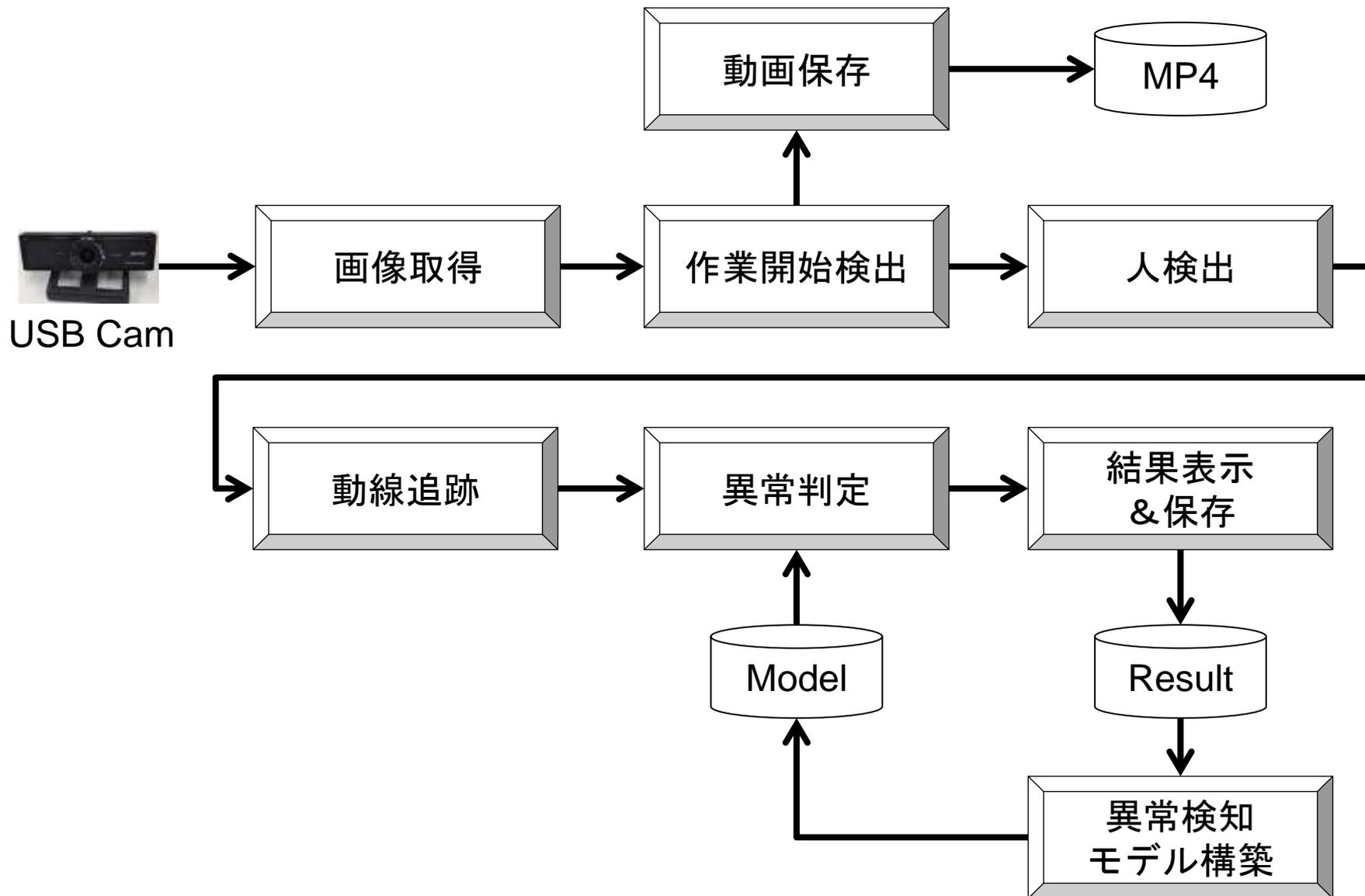
- 異常検知結果にもとづく当該工程でのフィードバック、テストなど次工程へのフィードフォワードにより、品質確認や修正作業を指示します。
- 場外不良の発生時にトレースバックし、作業起因の不良を切り分けることで、対応を迅速化します。
- 自社および顧客のグローバル製造拠点において、システムの有効性を評価しています。

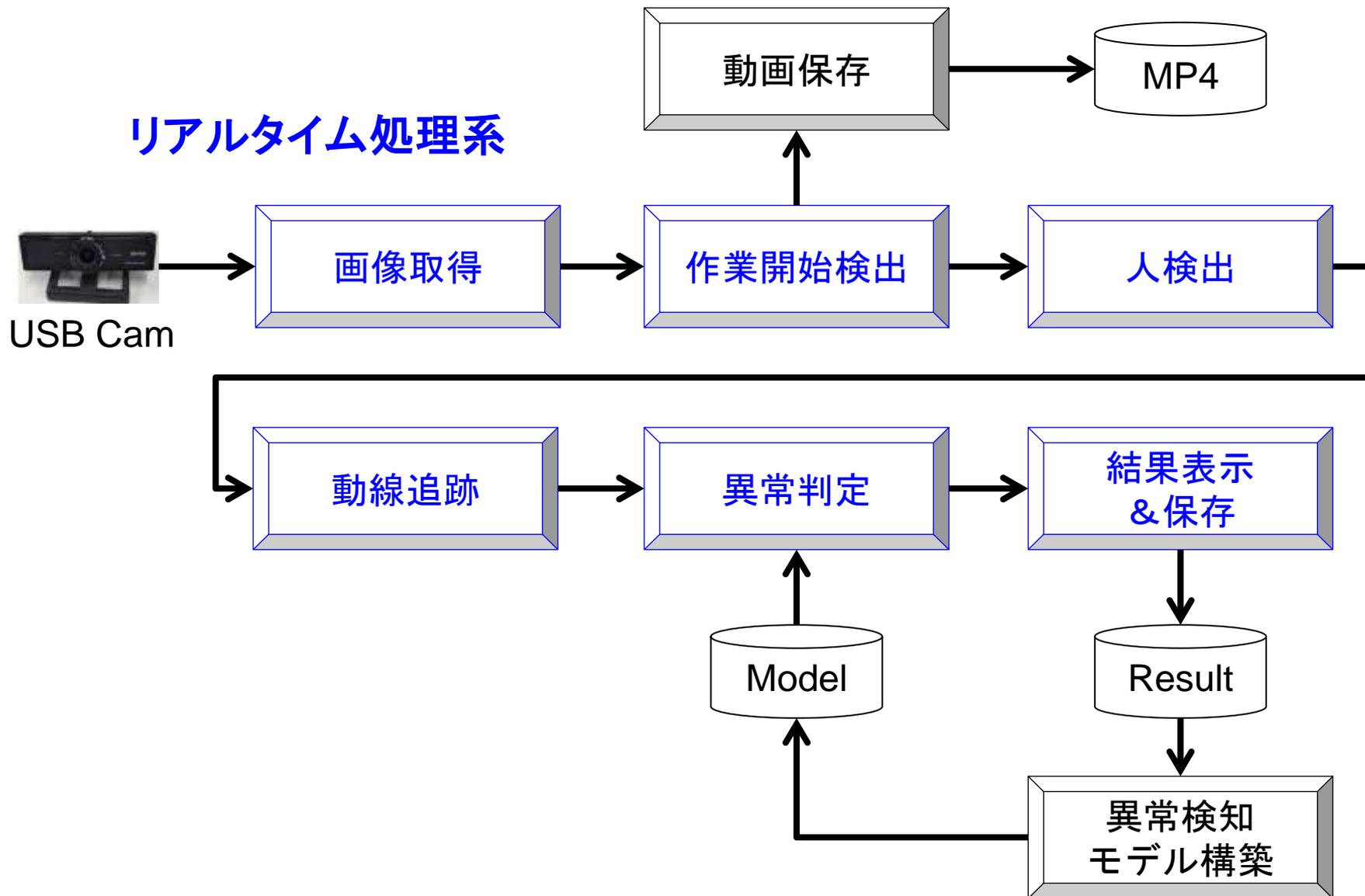




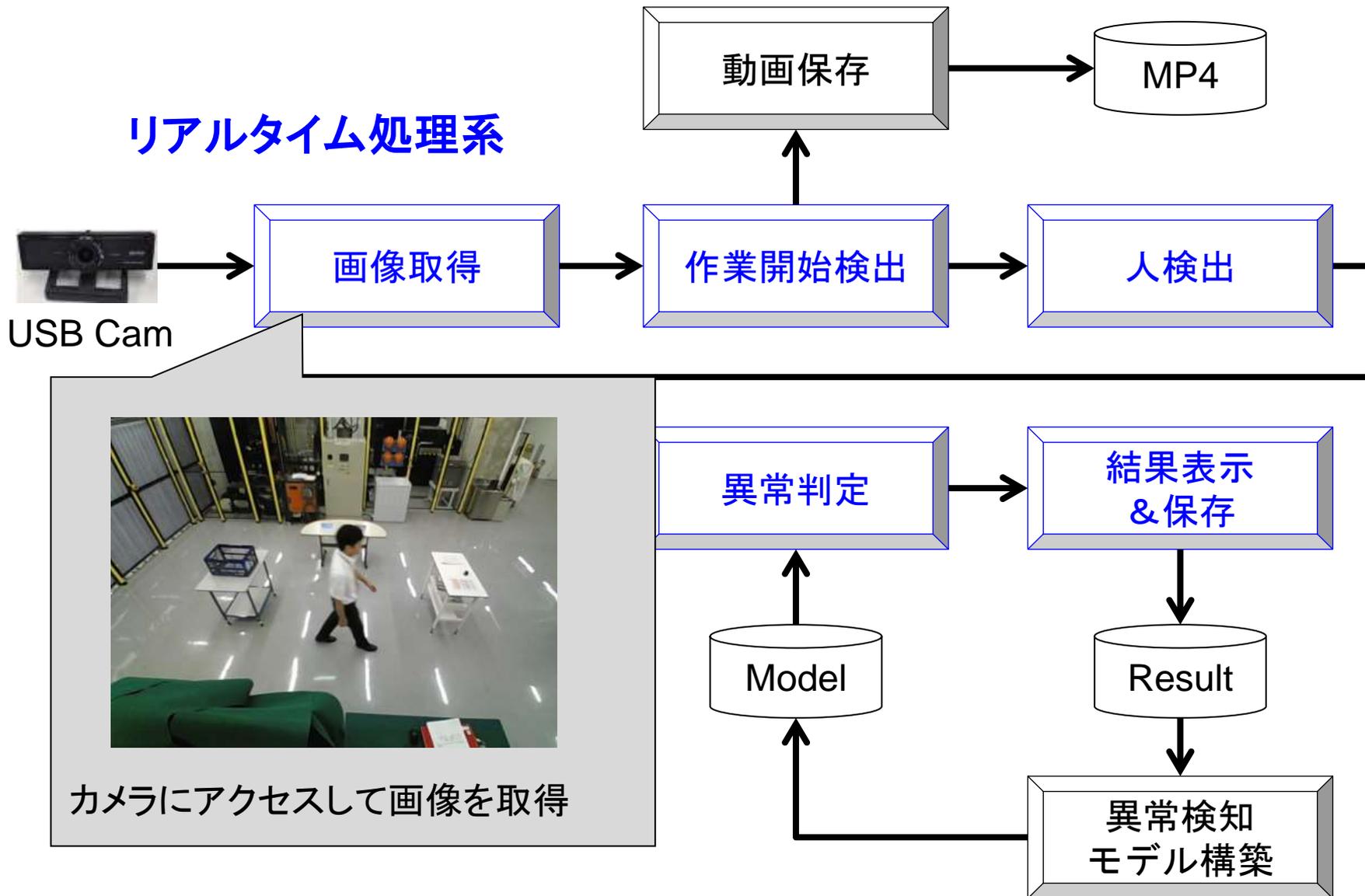


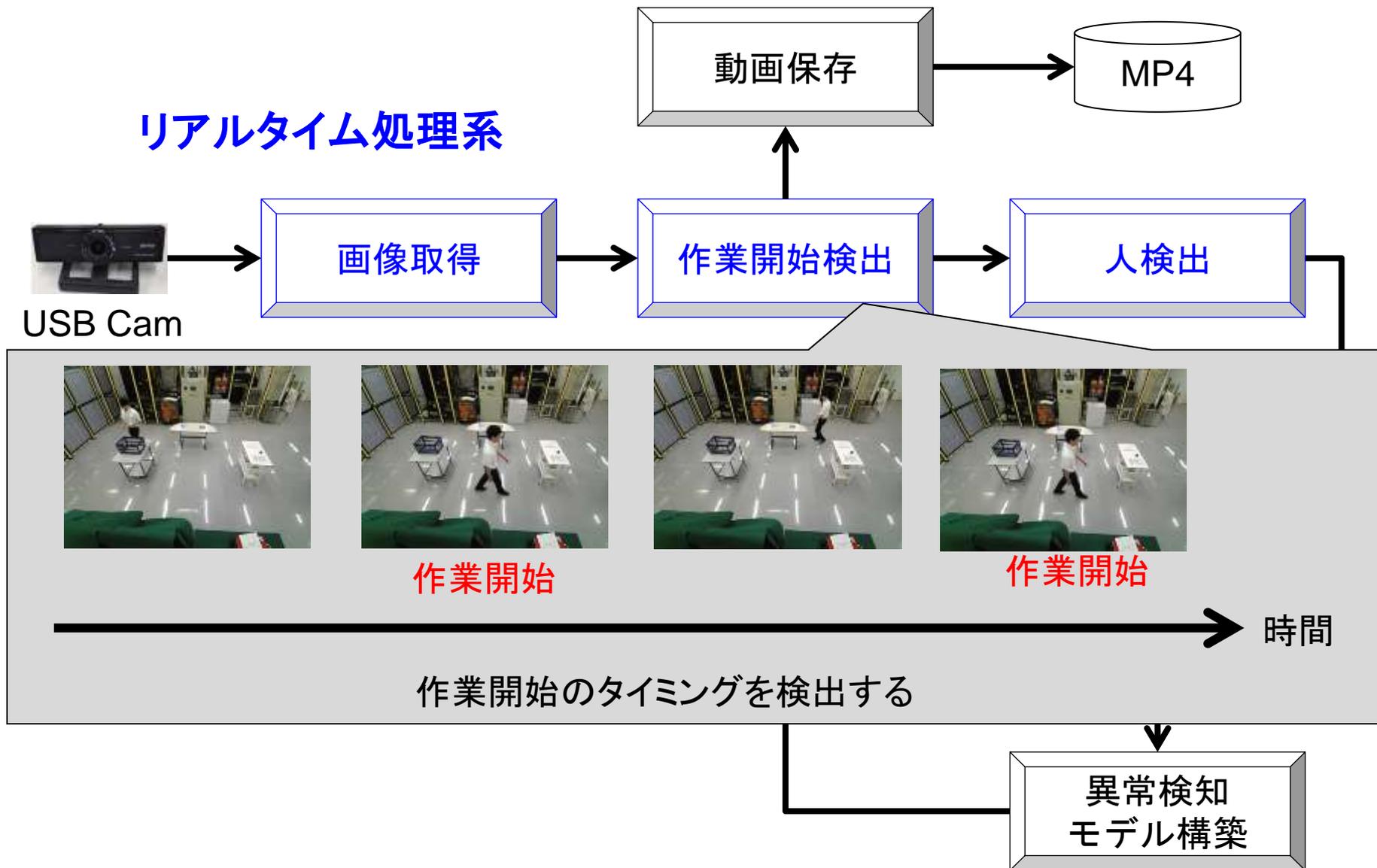




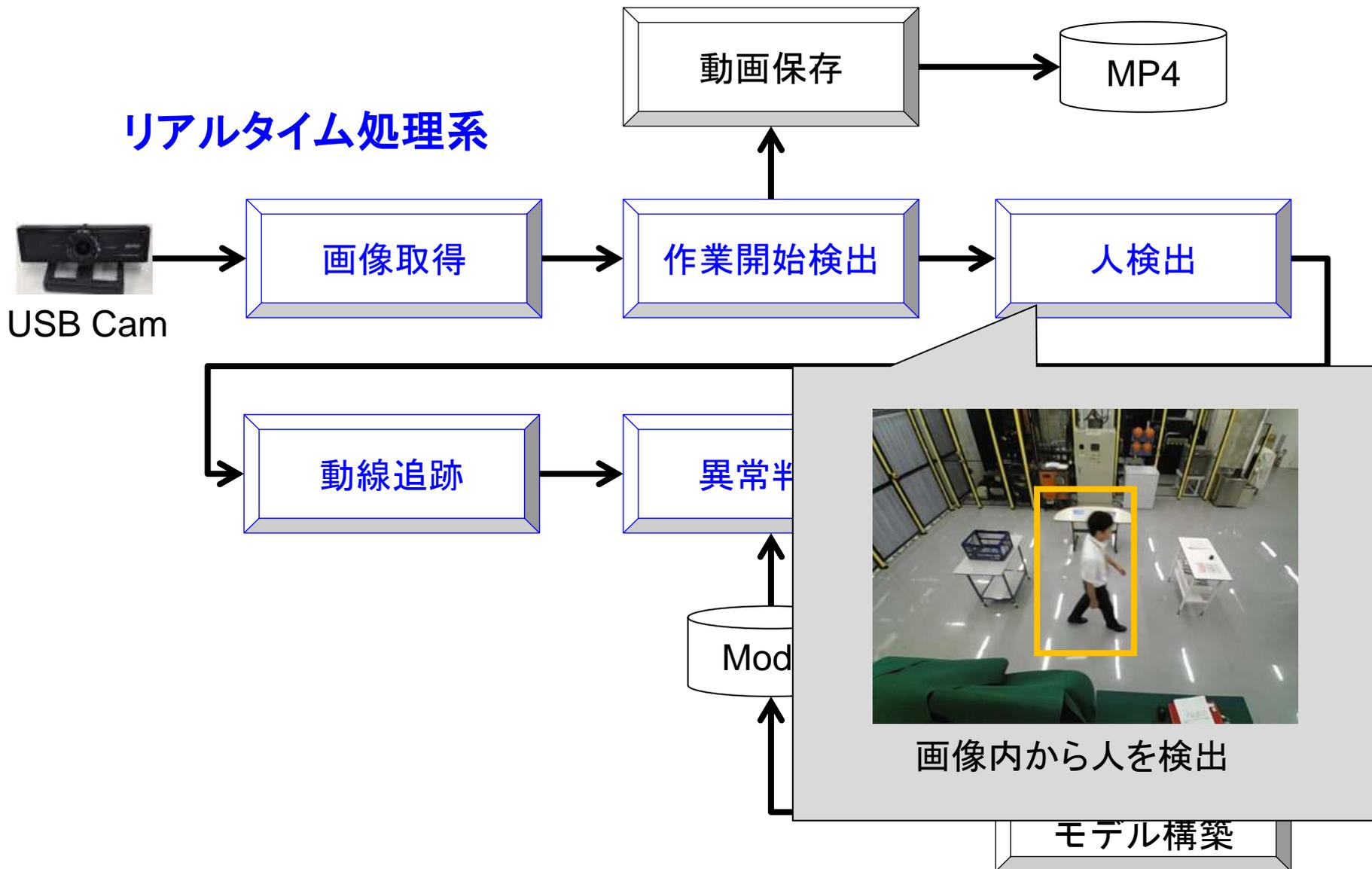


リアルタイム処理系

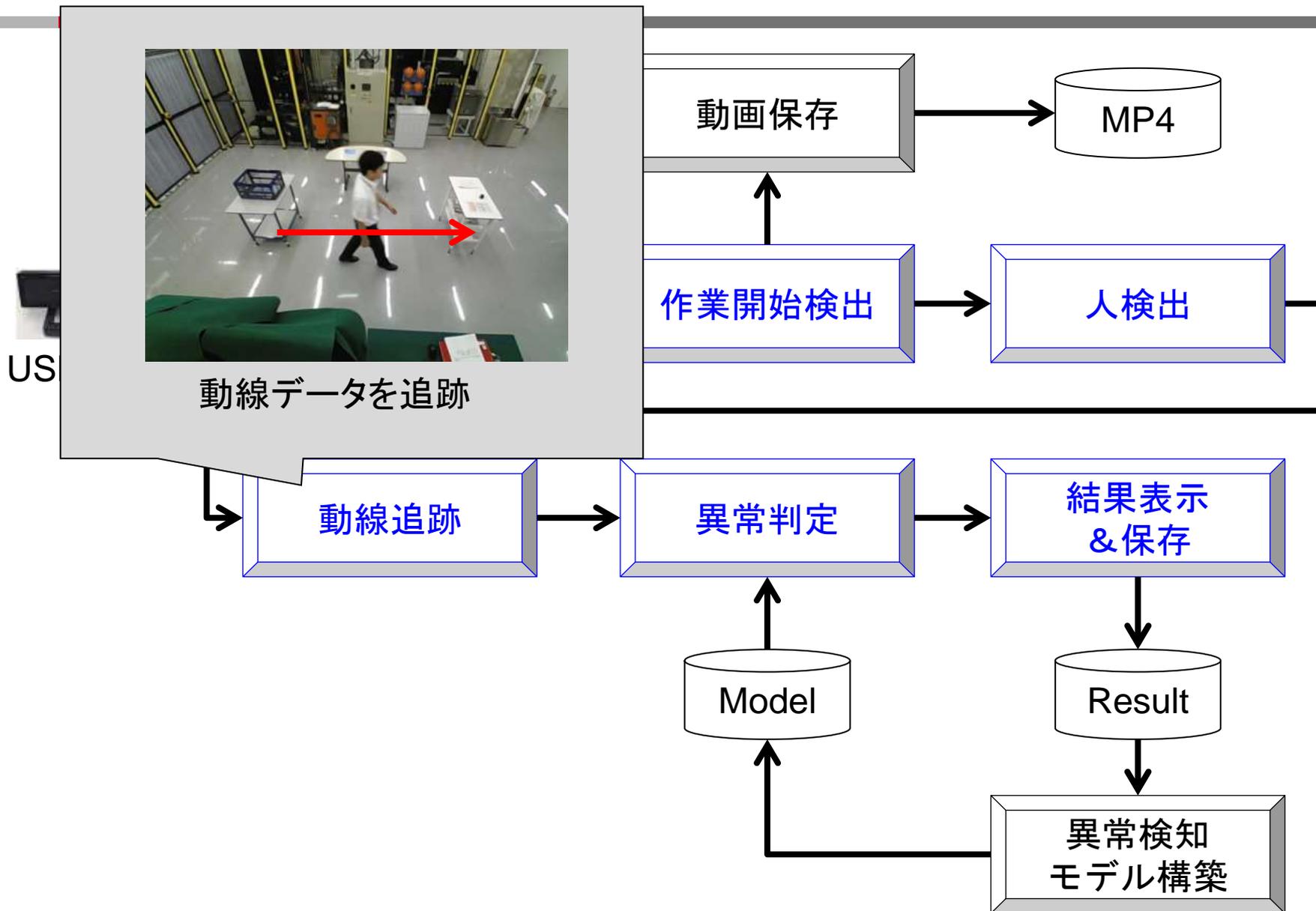




リアルタイム処理系

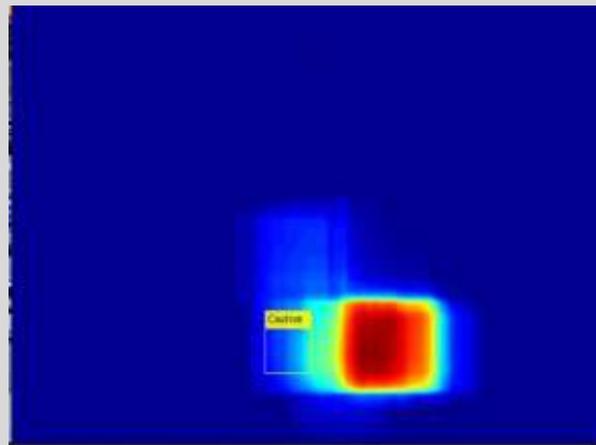
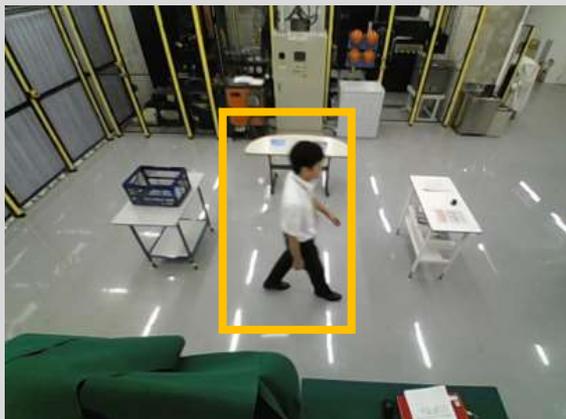


製造作業リアルタイム異常検知システム

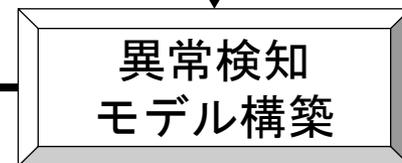
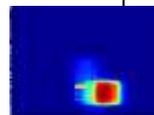
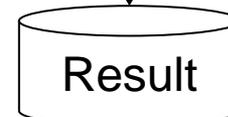
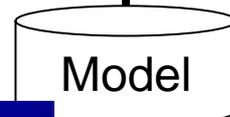
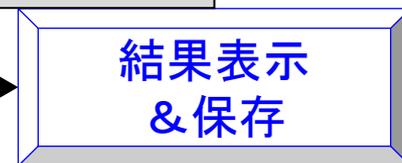
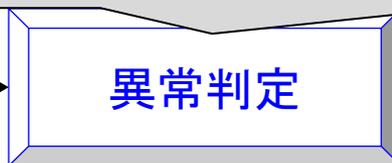


製造作業リアルタイム異常検知システム

US



ヒートマップと比較し, 異常検知



MATLAB EXPO 2016

MATLABによる作業異常検知システムの開発

1. 背景

2. 作業異常検知システムの概要

➤ 3. 開発の流れ

4. 応用例とまとめ

1 アルゴリズム評価用プロト

MATLAB m-File

目的: 技術開発 & 評価 (オフライン)

概要: 持ち帰ったデータを用いて, 研究所内にて, MATLAB上から, m-Fileを実行して, 評価.

2 PoC用簡易プロト

MATLABをコンパイルした
実行ファイル

目的: 技術評価 (オンライン)

概要: 配布可能な形でコンパイルした実行ファイルをオンサイトにインストールし, 評価.

3 PoC用プロト

C++開発のシステムと
MATLAB Coder DLL

目的: ユーザビリティ評価 (オンライン) & 製品仕様検討

概要: C++で開発したシステムから, MATLABのDLLを呼び出す形で実装. システムの使い勝手や, 製品仕様を検討.

4 製品開発

C++開発のシステムと
MATLAB Coder DLL

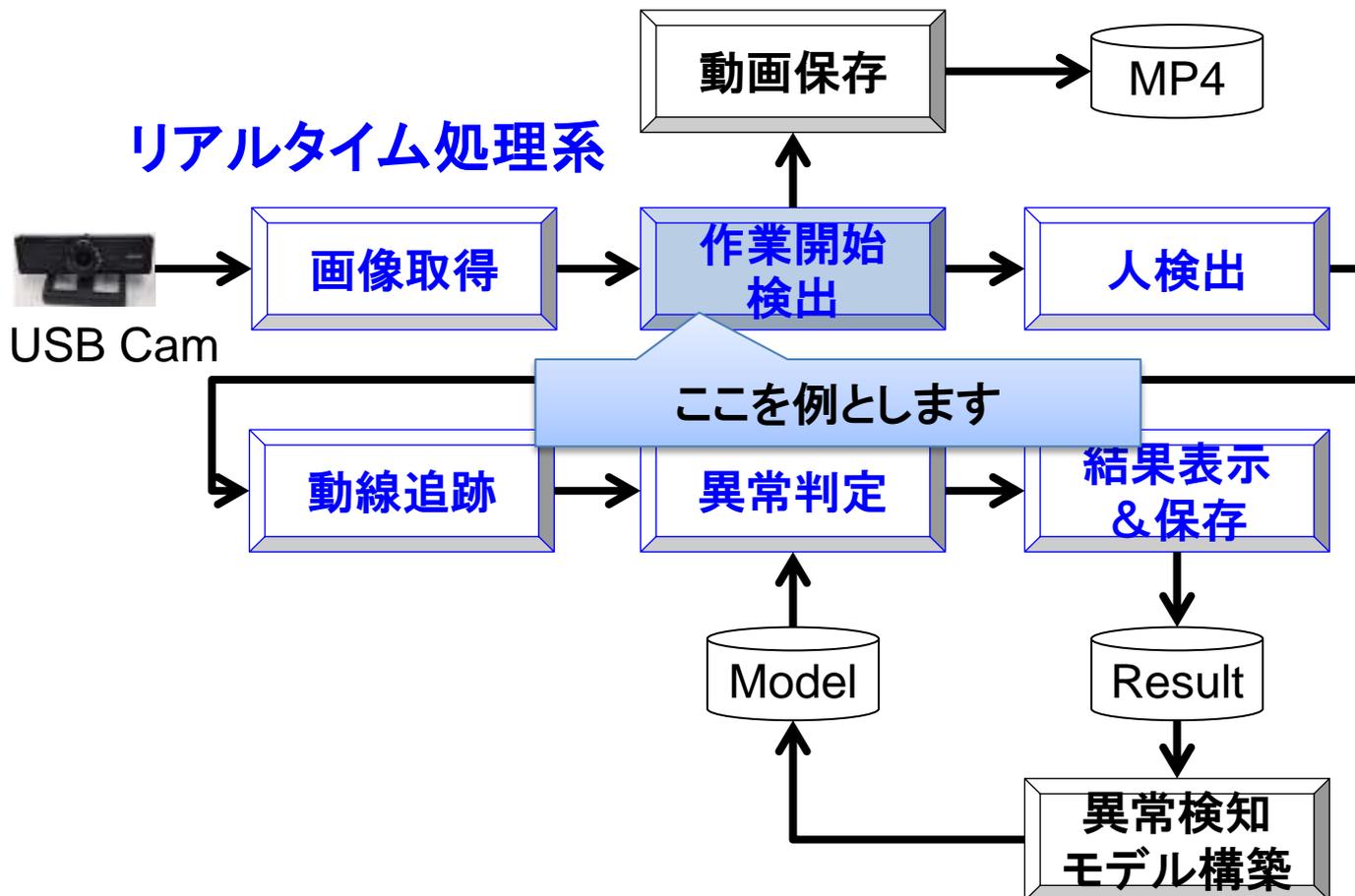
目的: 製品開発

概要: 3. で開発したプロトシステムの課題を解消する形で, 製品化開発を実施. 同時に, 品証テストも実施.

1 アルゴリズム評価用プロト

MATLAB m-File

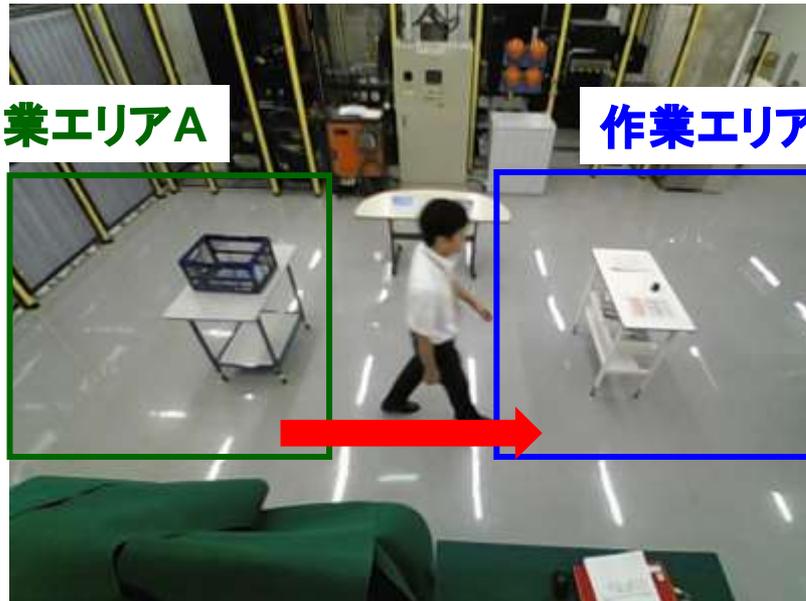
目的: 技術開発 & 評価 (オフライン)
概要: 持ち帰ったデータを用いて, 研究所内にて, MATLAB上から, m-Fileを実行して, 評価.



作業開始検出

作業エリアA

作業エリアB

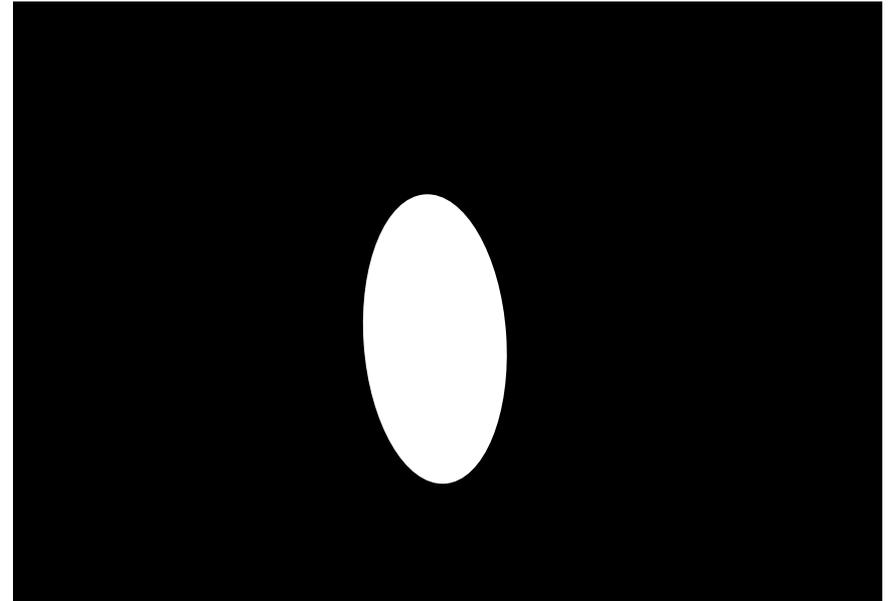


作業者が、作業エリアAから、作業エリアBに移動するタイミングを探すことを考えます。

物体の通過(移動)の検出の基本的な考え方
物体が通過するとき, 前景が画像内を移動するはず



画像

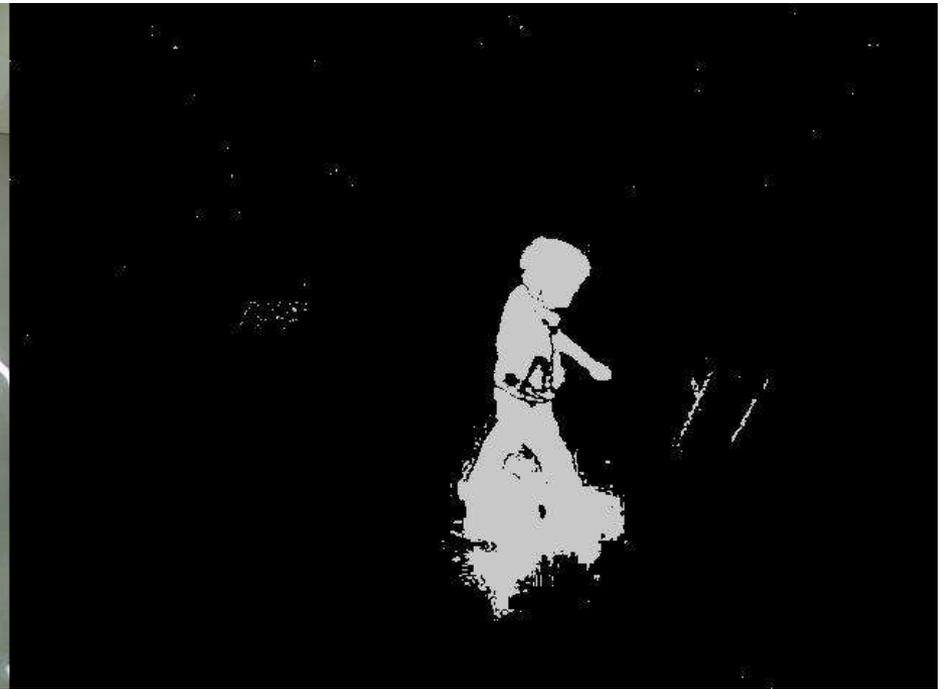


前景領域(白:前景, 黒:背景)

物体の通過(移動)の検出の基本的な考え方
物体が通過するとき, 前景が画像内を移動するはず



画像



前景領域(白:前景, 黒:背景)

物体の通過(移動)の検出の基本的な考え方
物体が通過するとき, 前景が画像内を移動するはず
→ 前景の面積の変化を元に, 検出する

作業エリアA
(From領域)

作業エリアB
(To領域)

To前景レート

To領域(青枠)における, 前景(白点)の割合

From前景レート

From領域(緑枠)における, 前景(白点)の割合

を考える.

物体の通過(移動)の検出の基本的な考え方
物体が通過するとき, 前景が画像内を移動するはず
→ 前景の面積の変化を元に, 検出する

作業エリアA
(From領域)

作業エリアB
(To領域)

To前景レート

To領域(青枠)における, 前景(白点)の割合

From前景レート

From領域(緑枠)における, 前景(白点)の割合

を考える.

通過初期: From前景レート > To前景レート

通過後期: From前景レート < To前景レート

となるはず.

物体の通過(移動)の検出の基本的な考え方

物体が通過するとき, 前景が画像内を移動するはず

→ 前景の面積の変化を元に, 検出する

作業エリアA
(From領域)

作業エリアB
(To領域)

To前景レート

To領域(青枠)における, 前景(白点)の割合

From前景レート

From領域(緑枠)における, 前景(白点)の割合

を考える.

通過初期: From前景レート > To前景レート

通過後期: From前景レート < To前景レート

となるはず.

物体の通過(移動)の検出の基本的な考え方
物体が通過するとき, 前景が画像内を移動するはず
→ 前景の面積の変化を元に, 検出する

作業エリアA
(From領域)

作業エリアB
(To領域)

To前景レート

To領域(青枠)における, 前景(白点)の割合

From前景レート

From領域(緑枠)における, 前景(白点)の割合

を考える.

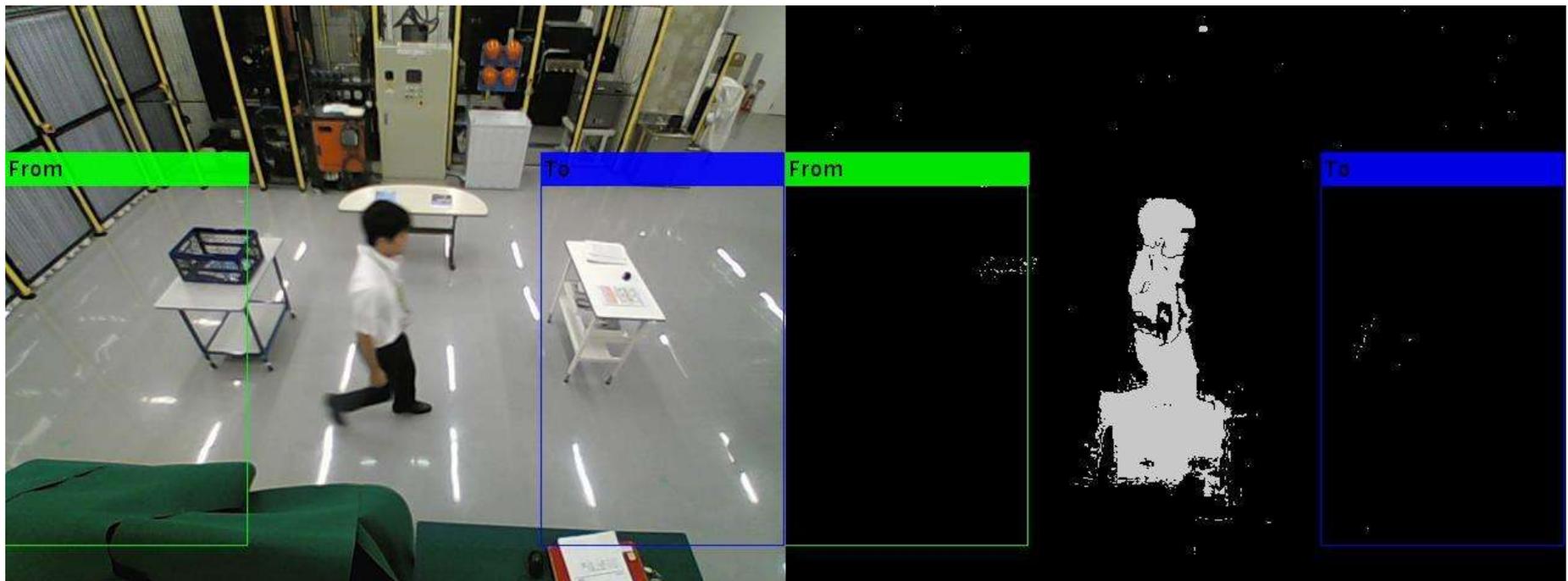
通過初期: From前景レート > To前景レート

通過後期: From前景レート < To前景レート

となるはず.

ということは,

検出統計量 := To前景レート - From前景レート
が, 増加する瞬間を探せば良い.



1 アルゴリズム評価用プロト

MATLAB m-File

2 PoC用簡易プロト

3 PoC用プロト

C++開発のシステムと
MATLAB Coder DLL

4 製品開発

C++開発のシステムと
MATLAB Coder DLL

1 既存アルゴリズム評価 Ph.1

大きく振るステップ

2 既存アルゴリズム評価 Ph.2

細かく振るステップ

3 問題抽出

4 アイデア立案

5 コーディング & 評価



1 既存アルゴリズム評価 Ph.1
大きく振るステップ

2 既存アルゴリズム評価 Ph.2
細かく振るステップ

3 問題抽出

4 アイデア立案

5 コーディング&評価

案1) 画像から作業者を認識し、
作業者の移動を追跡

案2) 動いている物体の移動状況を
追跡

案3) 作業開始時点での画像を記憶し、
類似の画像が発生したタイミング
を検出

ここでは、案2が最も優れていたとします。
※詳細は割愛します。

1 既存アルゴリズム評価 Ph.1

大きく振るステップ

2 既存アルゴリズム評価 Ph.2

細かく振るステップ

3 問題抽出

4 アイデア立案

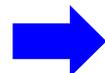
5 コーディング&評価

動いている物体の探し方

案1) 前景-背景分離

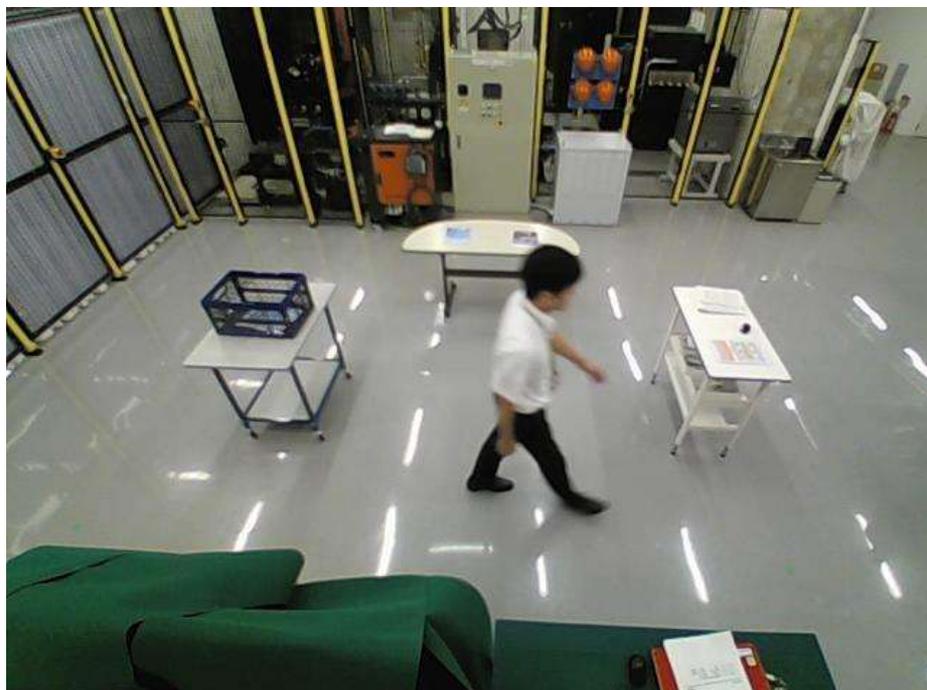
案2) オプティカルフロー

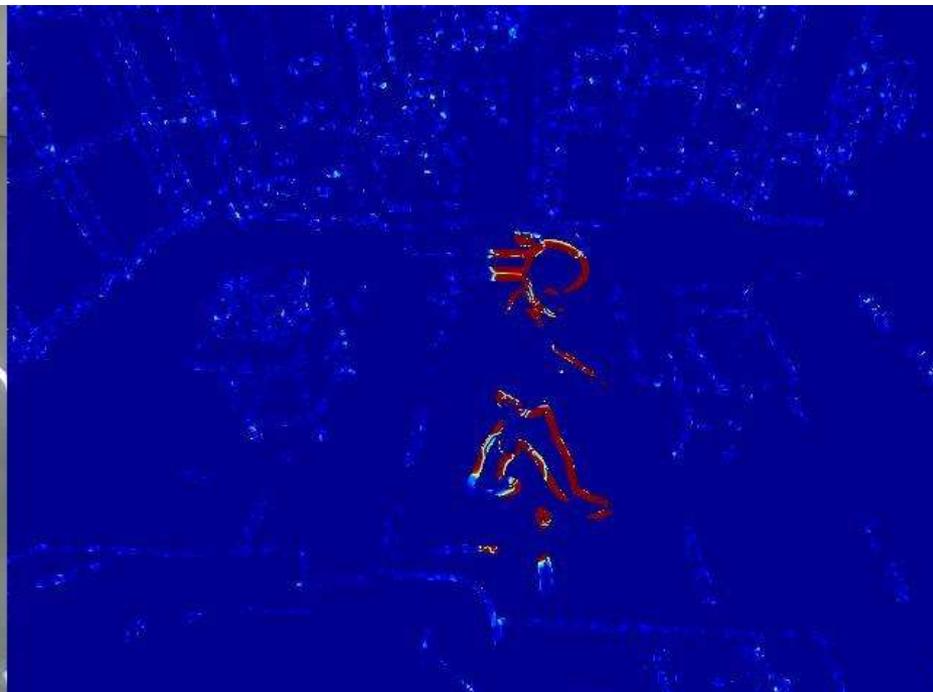
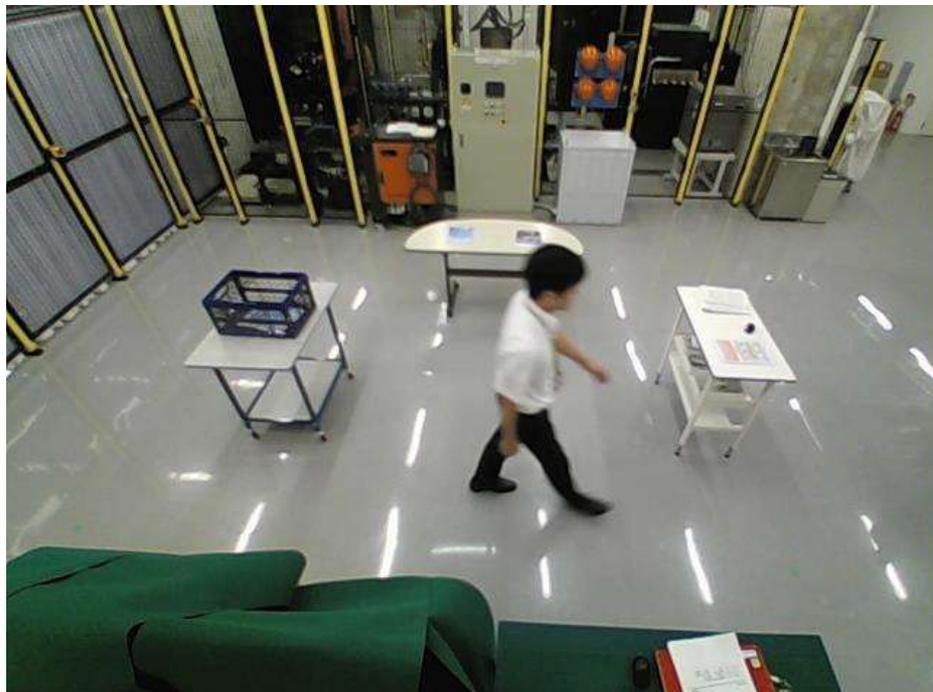
例えば、前景-背景分離を自力で実装しようとすると、「Adaptive background mixture models for real-time tracking」という論文に記載されているアルゴリズムを実装することになる。



イチから実装すると時間を要す
(MATLABの活用による短縮が可能)

評価映像(前景-背景分離)





1 既存アルゴリズム評価 Ph.1

大きく振るステップ

2 既存アルゴリズム評価 Ph.2

細かく振るステップ

3 問題抽出

4 アイデア立案

5 コーディング&評価

以下のステップでは、目標精度に未達であった場合、「問題抽出」、「アイデア立案」、「コーディング&評価」を繰り返す。



1 アルゴリズム評価用プロト

MATLAB m-File

目的: 技術開発 & 評価 (オフライン)

課題

- ・1. では、手作業で行っていた細かい処理の実装要
- ・各m-Fileを呼び出すフローコントロールの実装要。
- ・MATLAB Compilerの対応範囲に機能が限定される

概要: 配布可能な形でコンパイルした実行ファイルをオンサイトにインストールし、評価。

2 PoC用簡易プロト

MATLABをコンパイルした
実行ファイル

3 PoC用プロト

C++開発のシステムと
MATLAB Coder DLL

目的: ユーザビリティ評価 (オンライン) & 製品仕様検討
概要: C++で開発したシステムから、MATLABのDLLを呼び出す形で実装。システムの使い勝手や、製品仕様を検討。

4 製品開発

C++開発のシステムと
MATLAB Coder DLL

目的: 製品開発

概要: 3. で開発したプロトシステムの課題を解消する形で、製品化開発を実施。同時に、品証テストも実施。

1 アルゴリズム評価用プロト

MATLAB m-File

目的: 技術開発&評価(オフライン)

概要: 持ち帰ったデータを用いて, 研究所内にて, MATLAB上から, m-Fileを実行して, 評価.

2 PoC用簡易プロト

MATLABをコンパイルした
実行ファイル

3 PoC用プロト

C++開発のシステムと
MATLAB Coder DLL

4 製品開発

C++開発のシステムと
MATLAB Coder DLL

課題

・MATLAB→C++化の工数要
(お金と時間を要する)

・MATLAB Coderの対応範囲に機能が限定される.
→ このタイミングでフルスクラッチでの開発を
余儀なくされることも...

呼び出す形で実装. システムの使い勝手や, 製品仕様
を検討.

目的: 製品開発

概要: 3. で開発したプロトシステムの課題を解消する形
で, 製品化開発を実施. 同時に, 品証テストも実施.

1 アルゴリズム評価用プロト

MATLAB m-File

目的: 技術開発 & 評価 (オフライン)

概要: 持ち帰ったデータを用いて, 研究所内にて, MATLAB上から, m-Fileを実行して, 評価.

2 PoC用簡易プロト

MATLABをコンパイルした
実行ファイル

目的: 技術評価 (オンライン)

概要: 配布可能な形でコンパイルした実行ファイルをオンサイトにインストールし, 評価.

3 PoC用プロト

C++開発のシステムと
MATLAB Coder DLL

目的: ユーザビリティ評価 (オンライン) & 製品仕様検討
課題

- ・品証テスト期間を要する (≒ 開発期間)
- ・品証テストのためのアルゴリズムの詳細な説明が求められる.
- ・MATLABを用いること自体の是非を問われる.

4 製品開発

C++開発のシステムと
MATLAB Coder DLL

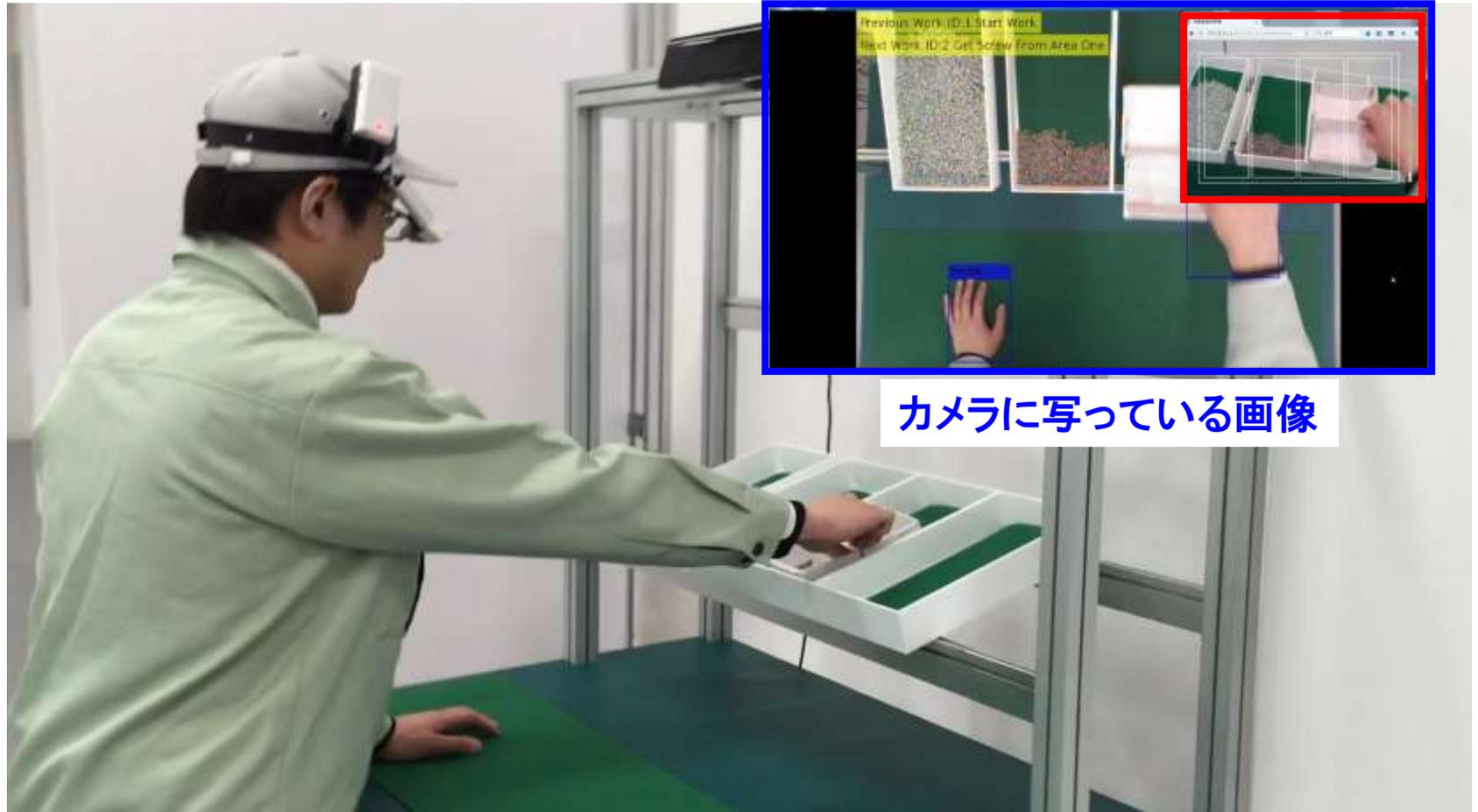
概要: 製品化開発を実施. 同時に, 品証テストも実施.

MATLAB EXPO 2016

MATLABによる作業異常検知システムの開発

1. 背景
2. 作業異常検知システムの概要
3. 開発の流れ
- 4. 応用例とまとめ

HMDで作業者が見ている映像



カメラに写っている画像

グローバル製造拠点における品質バラツキの解消、作業起因の場外不良の低減、品質トレーサビリティ確保によるリコール対応迅速化を狙いとし、作業異常検知システムを開発した。

開発システムは、カメラ画像から、作業の異常検知を行う。

異常検知の流れは、作業開始の検出、人の検出、動線の追跡、異常判定からなる。

アルゴリズムの開発から、実装までMATLABを活用しており、その流れを紹介した。

HITACHI
Inspire the Next