

# いまからはじめる MATLABによる画像処理・コンピュータービジョン2017

MathWorks Japan  
アプリケーションエンジニアリング部 (信号処理・通信)  
アプリケーションエンジニア  
福本 拓司

# “むずかしい”や“大変”を簡単に！ サポートするアプリ群

## R2017a

### レジストレーション推定



2枚の画像の位置合わせ

マウス操作で精度の良い手法を探すことが可能

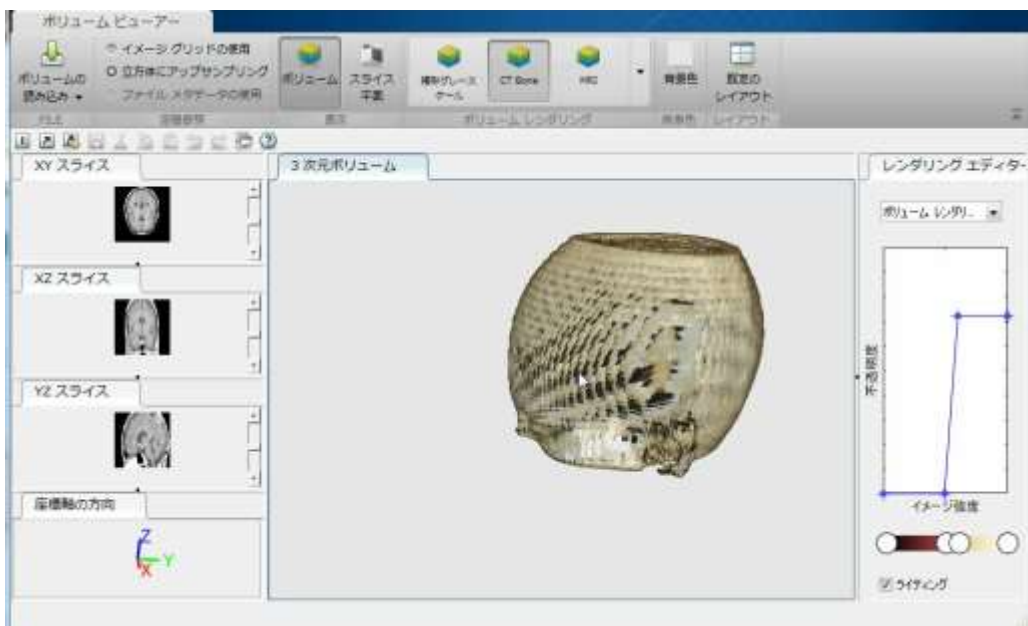
画像処理系だけで16種類以上のアプリ



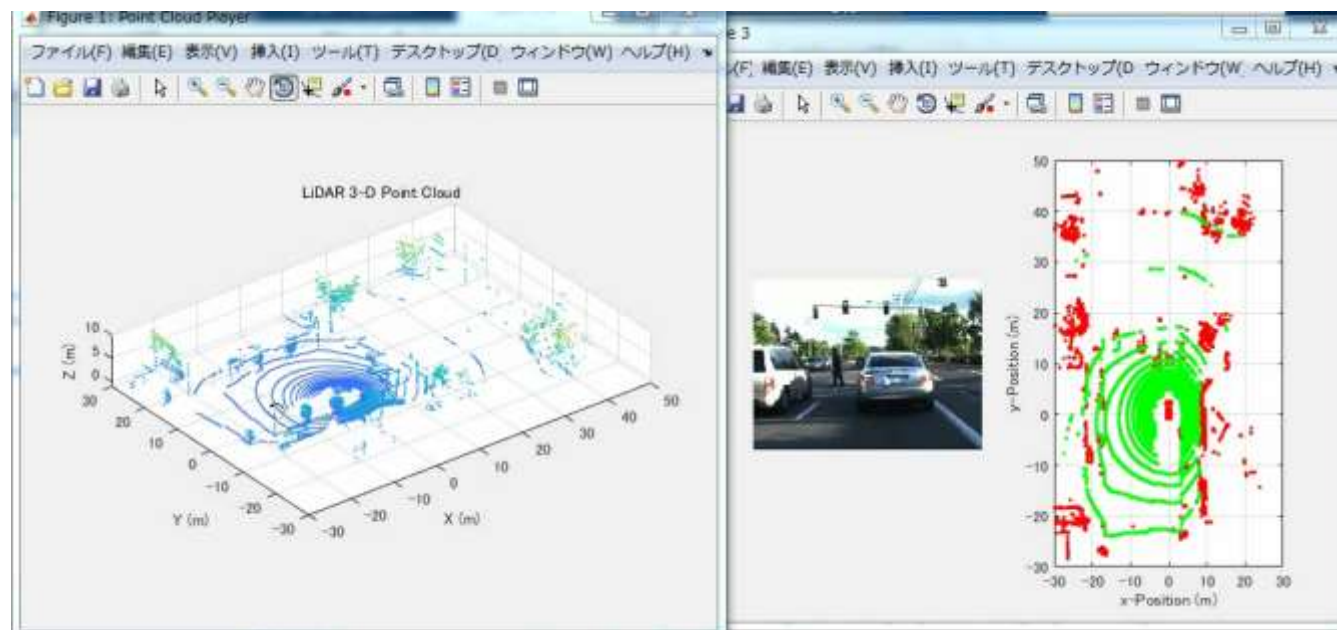
# 解析機能の拡大：3次元解析への対応

## R2017a

### ボリュームビューアー（3次元画像の解析）



### 3次元点群処理/可視化



# 応用領域の拡大

## R2017a

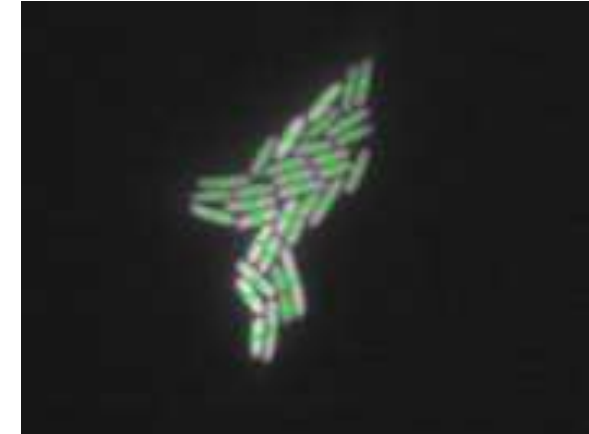
ディープラーニング&  
ロボットPID制御



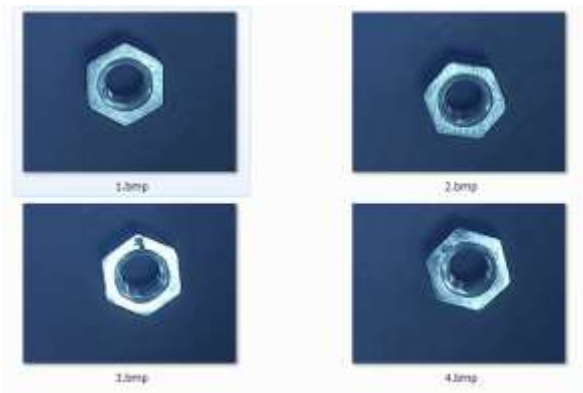
自動運転



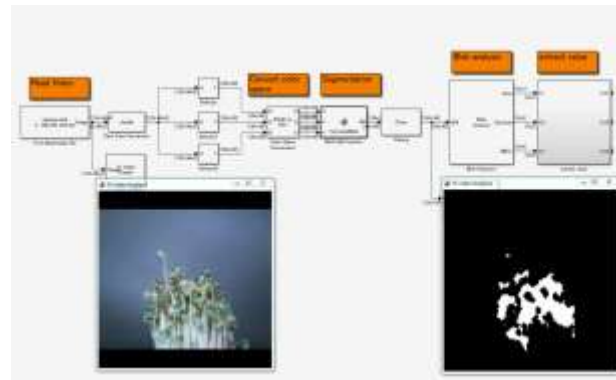
医用画像



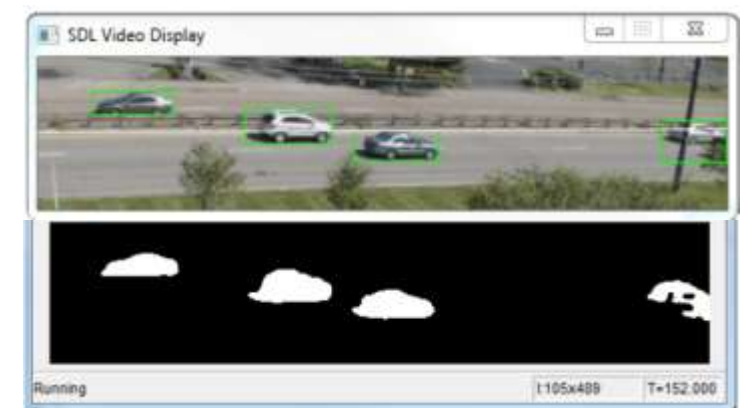
工業製品異常検知



農水産業



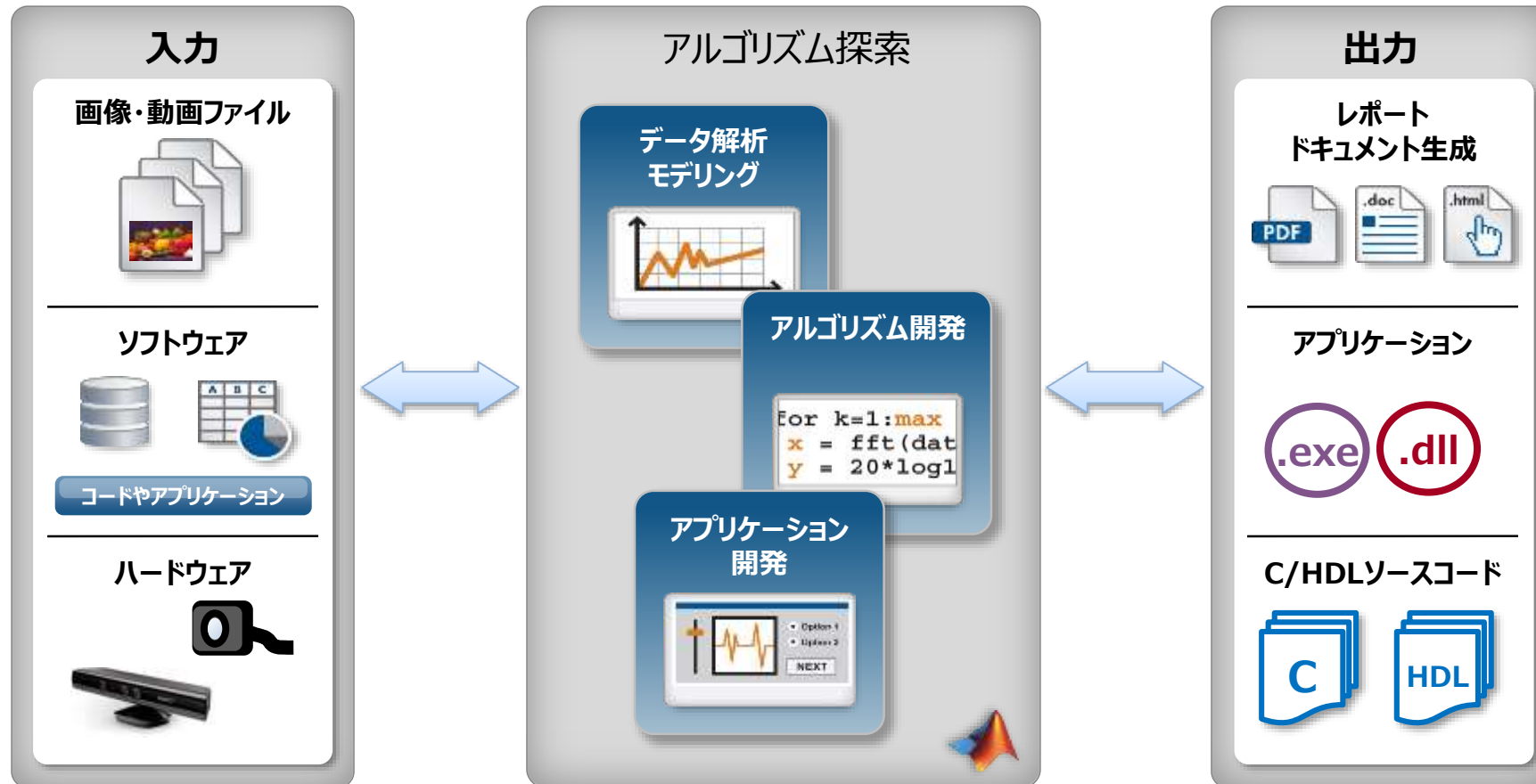
サービス(交通量計測)



# アジェンダ

- いまからはじめる画像処理
  - メーター値読み取りにおける画像処理ワークフロー
  - 統合的な開発環境であるMATLAB®
- 今話題！コンピュータービジョンの世界
  - コンピュータービジョンとは？
  - 人検出が1行？MATLABでおこなう人の行動解析のワークフロー

# MATLABによる効率的な画像処理開発フロー



- インタプリターによる、容易なパラメータ調整
- 高度で豊富な画像処理関数群
- アプリとコードの連携

# 各種画像/動画データフォーマットの読み込み・書き込みの例



## 画像の読み込み・書き込みのための関数

<b>imread()</b>	(MATLAB 基本関数)	グラフィックス ファイルからイメージを読み込み ( <b>bmp,gif,jpg,png,tif,..</b> )
<b>imwrite()</b>	(MATLAB 基本関数)	イメージをグラフィックス ファイルに書き込み (bmp,gif,jpg,png,tif,..)
<b>dicomread()</b>		<b>DICOM</b> イメージの読み込み
<b>dicomwrite()</b>		イメージをDICOMファイルとして書き込み
<b>nitfread()</b>		NITF(National Imagery Transmission Format)ファイルの読み込み
<b>hdrread()</b>		ハイダイナミックレンジ(HDR)イメージの読み込み
<b>hdrwrite()</b>		Radiance形式ハイダイナミックレンジ(HDR)イメージ ファイルの書き込み

## 動画の読み込み・書き込みのための関数

<b>vision.VideoFileReader()</b>	動画ファイルの読み込み(.avi , .mpeg,mp4, .m4v, .wmv,...)
<b>vision.VideoFileWriter()</b>	動画ファイルの書き込み

# 各種カメラからの画像データ直接取込み

## Image Acquisition Toolbox™



- 業界標準のHWからの動画取込み機能を提供
  - フレームグラバ (画像入力ボード)
  - DCAM 互換 FireWire (IIDC 1394)
  - GigE Vision
  - USB3 Vision
  - 一般的なUSB Webカメラ ←MATLAB本体でサポート
  - IPカメラ (MATLAB基本関数)
- Microsoft Kinect for Windows v1
- Microsoft Xbox One Kinect センサー
- カスタムアダプター開発キット
- Simulink ブロック



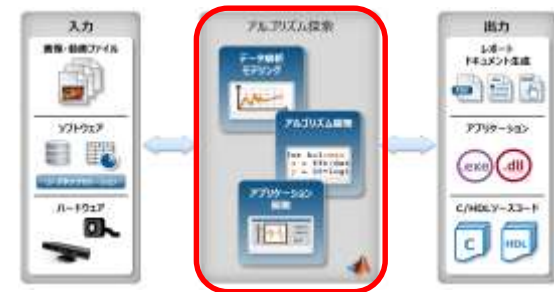
RGB画像 + 骨格座標



深度画像



# 画像処理による速度メーターの測定



デモ

画像処理

針を検出

目盛りの検出

解析 &  
直線化

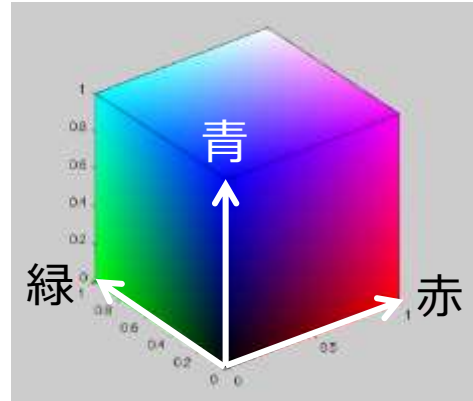
速度0に対する  
針の角度

ゴール

速度  
読み取り

# 色空間

## RGB色空間



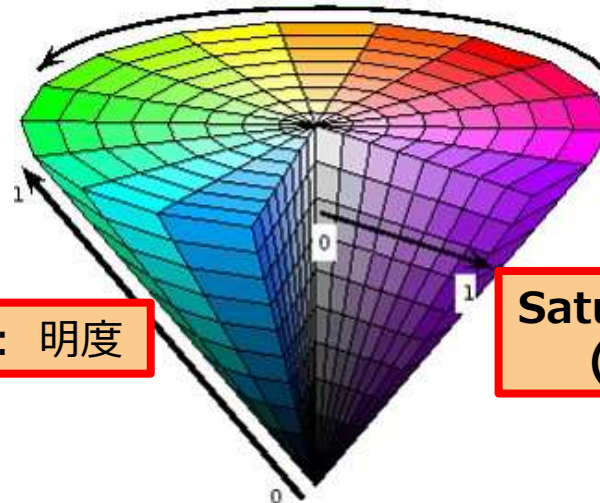
白	:	$(R,G,B) = (1, 1, 1)$
黄色	:	$(R,G,B) = (1, 1, 0)$
シアン (～水色)	:	$(R,G,B) = (0, 1, 1)$
緑	:	$(R,G,B) = (0, 1, 0)$
マゼンタ (紅紫色)	:	$(R,G,B) = (1, 0, 1)$
赤	:	$(R,G,B) = (1, 0, 0)$
青	:	$(R,G,B) = (0, 0, 1)$
黒	:	$(R,G,B) = (0, 0, 0)$

## HSV色空間

Hue : 色相  
 Saturation : 彩度 (0 : 白/灰色/黒)  
 Value : 明度

Hue : 色相

赤 : 0 / 1.0



Value : 明度

Saturation : 彩度  
(白/灰色/黒は0)

# 画像処理ワークフロー

色のしきい値アプリ  
による二値化

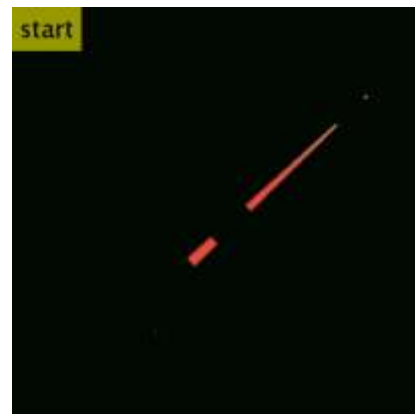


針を検出

目盛りの検出

マウス操作による  
基準位置の指定

モルフォロジー処理: bwmorph  
領域解析: regionprops



座標解析  
& 直線化

角度の解析



基準に対する  
針の角度  
読み取り

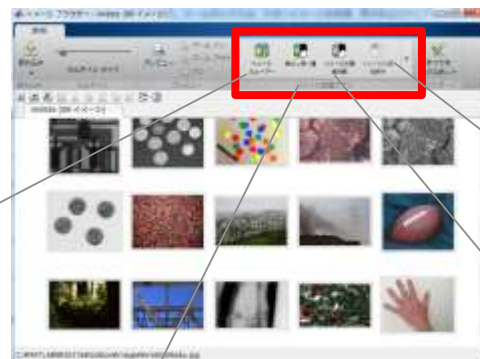
速度  
読み取り

129.2864 km/h

アプリ & 高度な関数 & マウス操作 で  
“手軽かつ柔軟”に画像処理による効率化を目指すことができます。

# はじめての画像処理を強力にサポート：画像処理用アプリケーション

直感的なGUI操作のアプリケーション

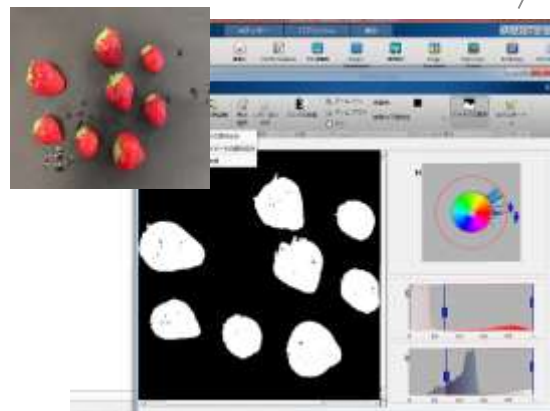


イメージブラウザー

イメージビューアー

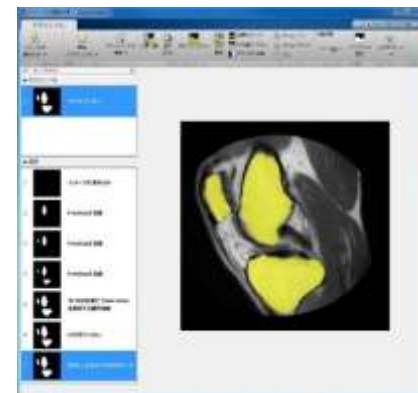


色のしきい値



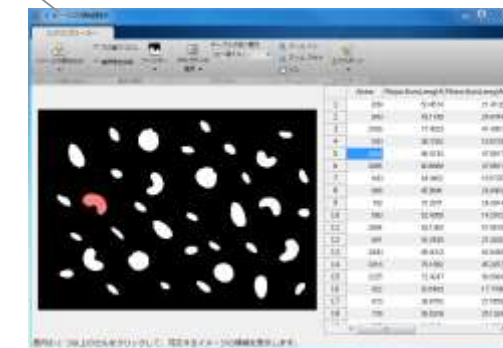
<http://jp.mathworks.com/videos/color-based-segmentation-using-the-color-threshold-app-117394.html>

イメージの領域分割



<http://jp.mathworks.com/videos/image-segmentation-application-120733.html>

イメージの領域解析



<http://jp.mathworks.com/videos/calculate-region-properties-using-image-region-analyzer-116911.html>



変数やMATLABコードとして出力可能

# プログラムに柔軟性を与える： GUIでのマウス操作



getpts()  
クリックしたpixel  
情報を取る

対話型ツールの構築  
40以上の関数

<https://www.mathworks.com/help/releases/R2016b/images/building-guis-with-modular-interactive-tools.html>

関心領域(ROI)の一部が欠けてしまっても。。。

Imfreehand() フリーハンド



囲んだ部分を追加



Imrect() 四角を描く



囲んだ範囲で  
輝度による2値化関数



マウス操作



使い慣れた関数



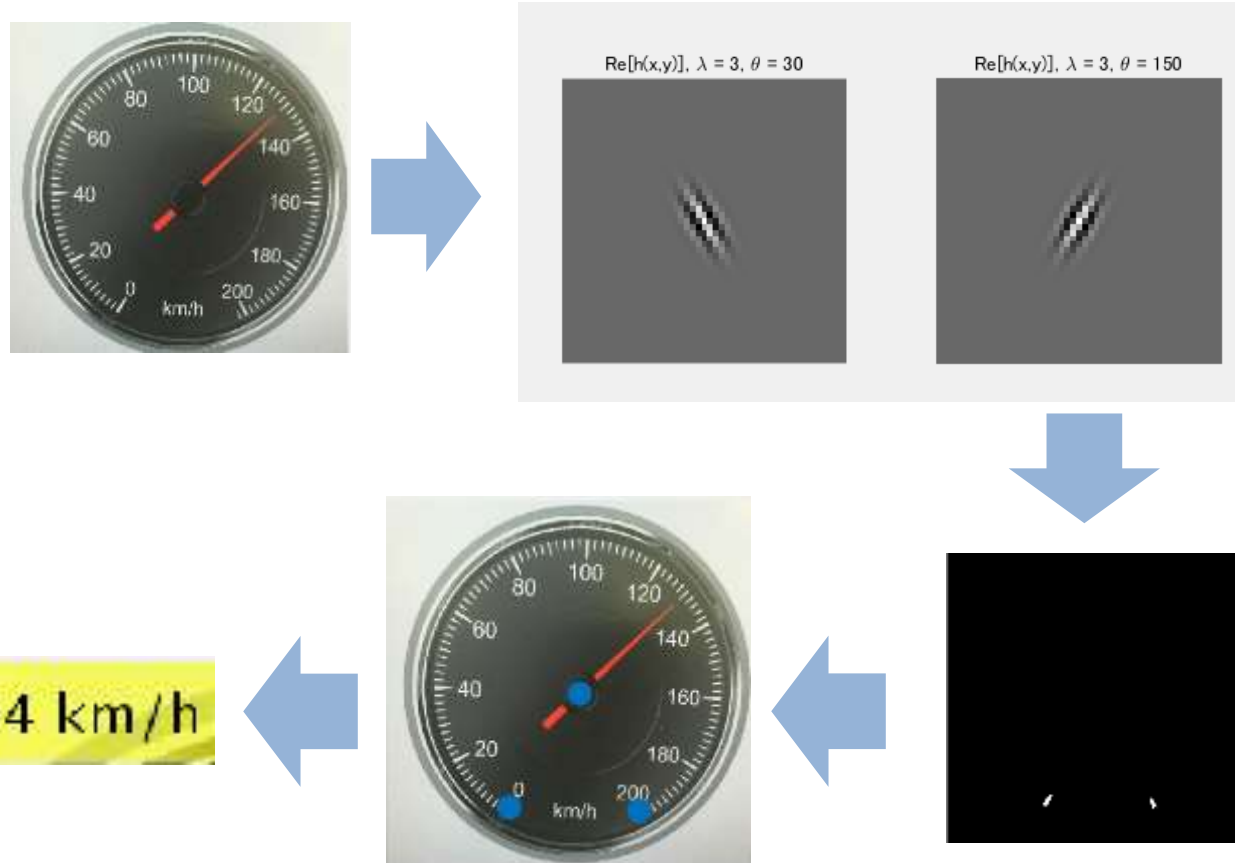
自分だけのアプリ

難しい処理に直面しても

- プロジェクトの成果物を柔軟に作ることができる
- 着実な作業効率向上を目指せる

# 目盛り検出の自動化と文字認識

ガボールフィルタと  
モルフォロジー処理で目盛り抽出



あとは同様の処理で  
角度→速度を測定

Centroidを測定し座標を利用

OCRを用いた数字の読み取り



ocr() 文字認識関数により  
目盛り数値読み取り

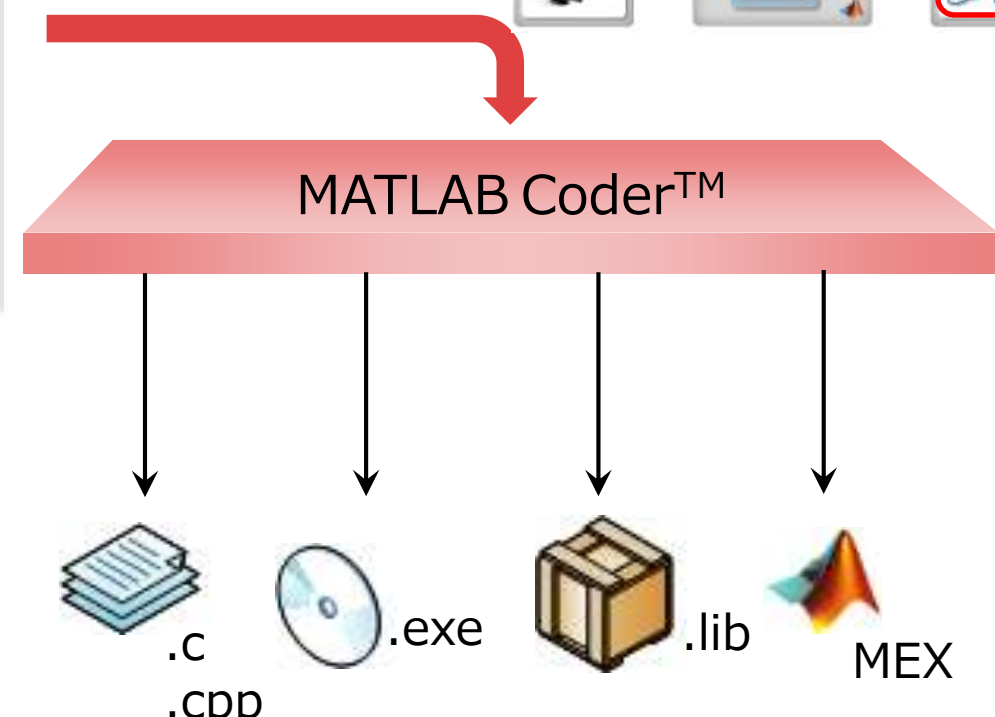
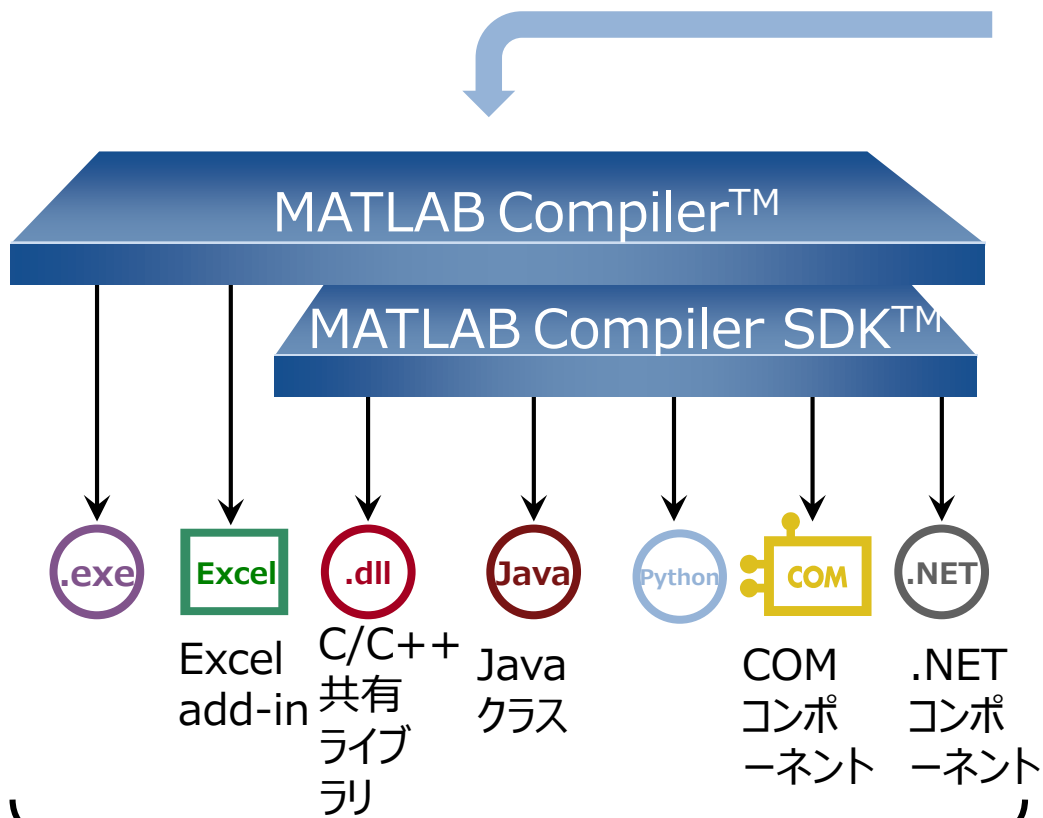
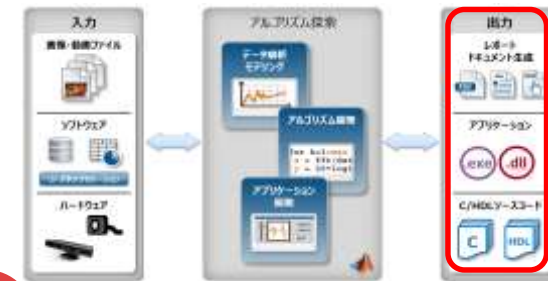
maxspeed =  
200


# アプリケーション/コード配布

MATLABファイル

```

1 function edge_img = detect_edge(img, threshold)
2
3
4 % グレースケール変換
5 img_gray = single(img) / 255;
6
7
8 % フィルタカーネル
9 k = [ 1 2 1; ...
10       0 0 0; ...
11       -1 -2 -1];
12
13 % たたき込み演算
14 H = conv2(img_gray, k, 'same');
15 V = conv2(img_gray, k, 'same');
16
17 % エッジ検出
18 E = sqrt(H.^2 + V.^2);
19 edge_img_bin = E > threshold;
20 edge_img = uint8(edge_img_bin) * 255;
    
```



 無償でダウンロードできるMCRで実行

C/C++言語での利用

作成したアルゴリズムをMATLAB環境以外で実行可能

# アジェンダ

- 手軽にはじめる画像処理
  - メーター値読み取りにおける画像処理ワークフロー
  - 統合的な開発環境であるMATLAB®
- 今話題！コンピュータービジョンの世界
  - コンピュータービジョンとは？
  - 人検出が1行？MATLABでおこなう人の行動解析のワークフロー



# コンピュータビジョンとは？



デモ

動画像からコンピュータでシーンの理解などを行う技術  
(物体の認識、トラッキング、距離の把握)



## 画像処理

ノイズ補正  
コントラスト補正  
測定  
...

## コンピュータビジョン

検出  
認識  
識別  
トラッキング  
...

## シーンの理解

道路  
人物  
自転車  
トラック  
アクシデント  
...

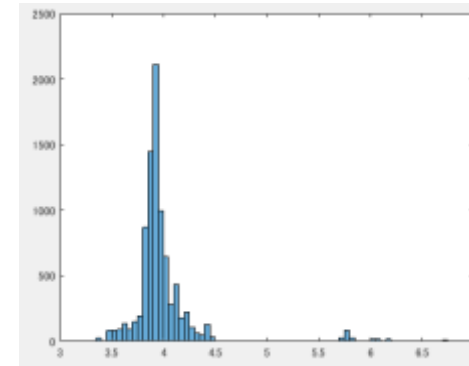
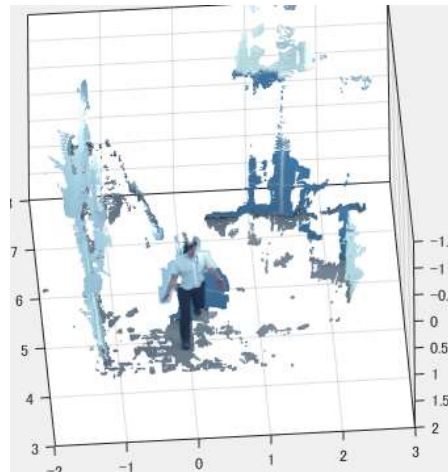
# コンピュータビジョンによる人の行動解析



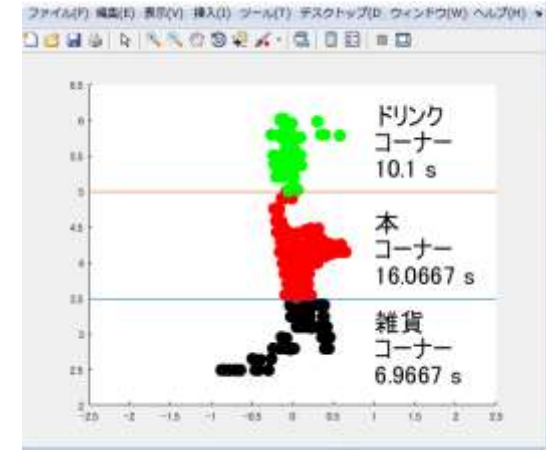
**R2016a**

`detectPeopleACF()`  
たった1行で人検出

(顔検出も1行で  
`vision.CascadeObjectDetector`)



人周辺の  
点群座標を解析



## ステレオビジョンワークフロー :

R2014a

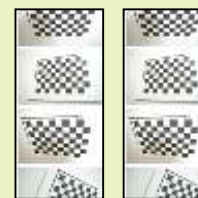
R2014b

## 事前 キャリブレーション

キャリブレーション用パターンの撮影



ステレオカメラ キャリブレーション



パラメータ

ステレオ画像の並行化



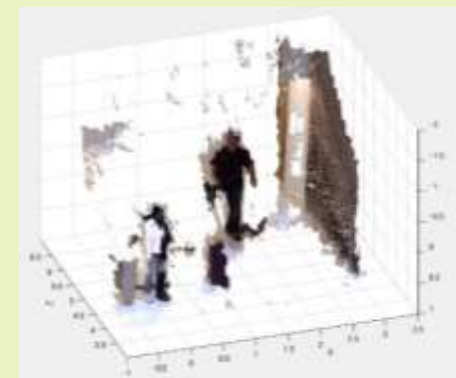
左右の画像のずれから、視差の計算



R2015a

処理速度の高速化

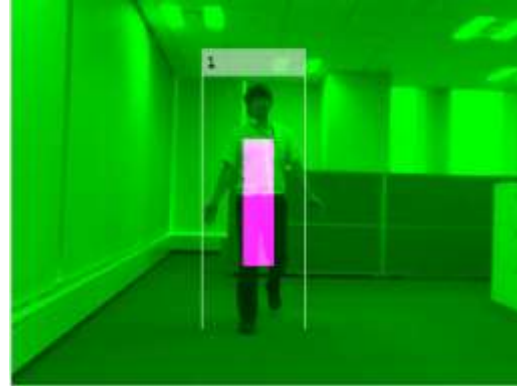
三次元空間の再構築



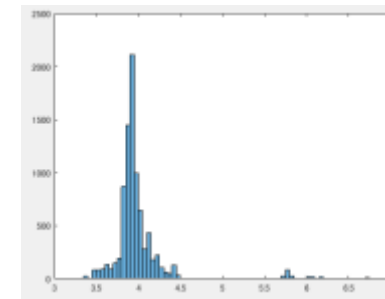
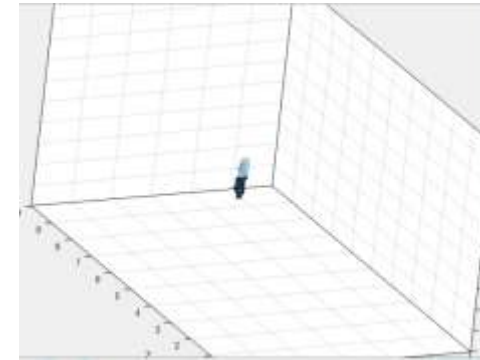
# 人認識アルゴリズムと3次元点群を用いた人の行動解析



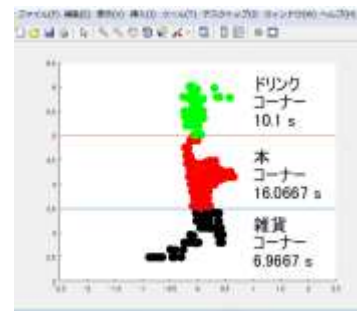
人検出のボックスを使って  
人周辺の視差画像だけを切り抜き



人周辺のみ3次元点群計算



点群情報の解析



行動解析結果  
プロット

- ・人認識 & 3次元点群で「各コーナー滞在時間」を解析
- ・ディープラーニングによる認識にも対応
- ・豊富な解析&可視化機能がパワフルにサポート

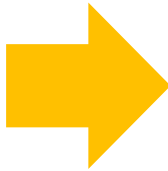
## 動きの認識 機能例

### 動きの認識

#### ● 動き検出



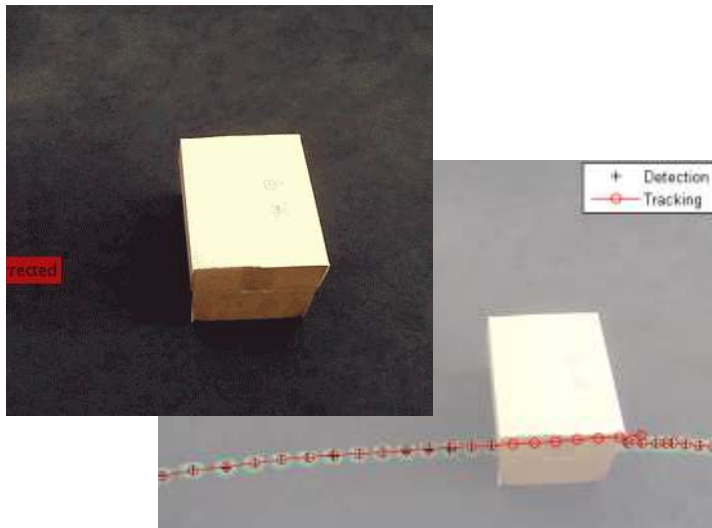
動いている領域の検出



車のカウント



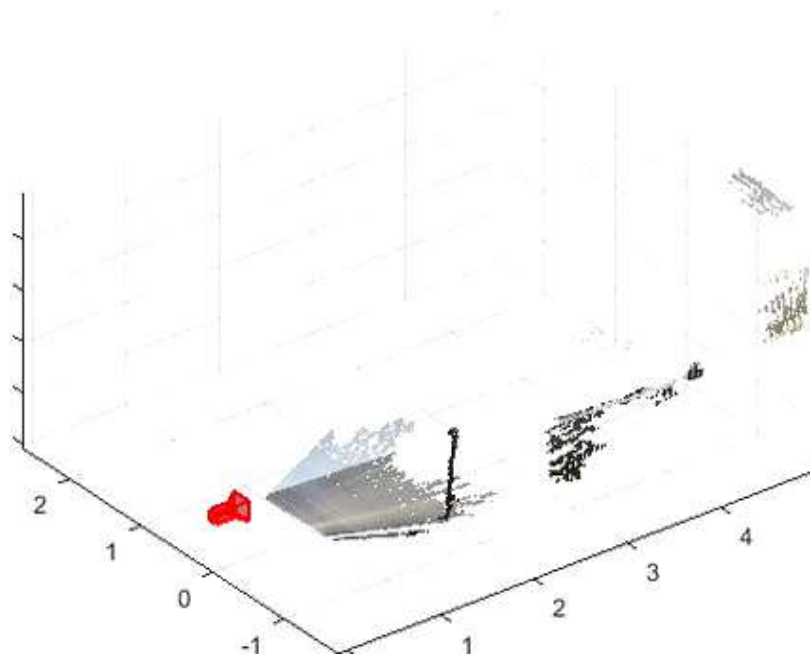
#### ● トラッキング（追跡）



隠れている物体の位置予測  
誤検出の低減  
個々の物体の認識（数量計測等）

## 3次元点群(ポイントクラウド)

3次元空間



- 距離測定
- 自己位置推定(SLAM)
- MAPを用いて行動の最適化

ポイントクラウド構築に必要なのは



深度(奥行き)

手法の例

- Kinect (深度センサ付)

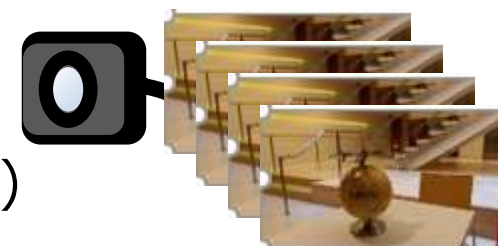


- ステレオビジョン



R2016a

- カメラ一台+複数枚画像  
(Structure From Motion)



# 高度な物体認識&解析を実現する豊富な関数群

- MATLAB®
- Image Processing Toolbox™
- Computer Vision System Toolbox™

## 物体認識

- 画像処理 (色・形 等)
- **人認識アルゴリズム**
- 背景差分法
- 画像特徴量
- 機械学習 (Statistics and Machine Learning Toolbox™)
- **ディープラーニング** (Neural Network Toolbox™他)

## トラッキング

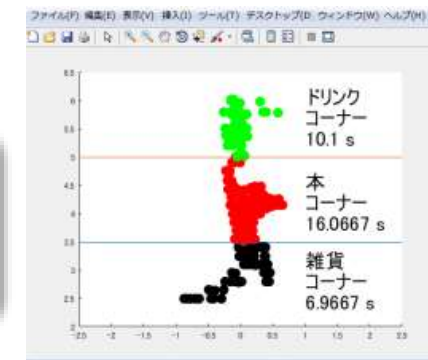
- **カルマンフィルタ**
- KLT
- パーティクルフィルタ (Robotics System Toolbox™)

## 3次元空間

- Kinect
- **ステレオビジョン**
- 単眼カメラSFM

## 解析

- MATLAB Toolbox各種



アプリケーションへの適用

**ロボット・ADAS 制御 等**

(Robotics System Toolbox™他)

## MATLABなら

認識から解析までを一貫して同じ環境。  
高度な機能を含むサンプルを用いてすぐにアイデアを試すことができます。

## まとめ

- **MATLABは  
画像取り込みからアプリケーション配布までできる開発環境**
- **アプリからのコード生成&高度な関数群により  
効率的で柔軟なアルゴリズム開発が可能**



**ご清聴ありがとうございました**