



Ai Robot Research Center



植物生体研究



動物生体研究



画像処理研究

画像処理を始めよう。

ROBO-ONEに適用の可能性を探る。

画像の分類において深層学習が人を超える能力を示し、第三次人工知能ブームを引き起こしている。この人工知能技術を扱ううえで画像処理の基本を理解することは不可欠である。人工知能ロボット研究所の事例をもとに解説する。

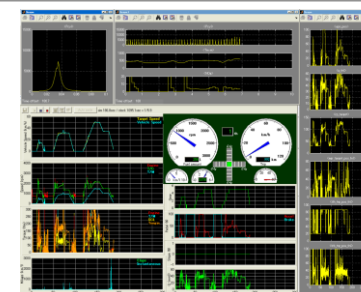
2018年7月7日

西村輝一

株式会社人工知能ロボット研究所 代表取締役

(株)人工知能ロボット研究所の基盤技術

商業車の
研究開発
(MBD)



映像

耐久信頼性の高い商用車の開発。様々な分野のロボットの現状を理解。

ロボットの
技術開発
(自律神経)



人工知能
技術開発
(画像処理)



ROBO-ONE関連書籍

ROBO-ONEのための二足歩行ロボット製作ガイド



二足歩行ロボットのモデルベース開発



ROBO-ONEで進化する二足歩行ロボットの造り方



ROBO-ONEにチャレンジ!二足歩行ロボット自作ガイド

World Robot Summit

経済産業省はオリンピックに合わせて、ロボットのオリンピック**World Robot Summit**を実施する。
World Robot Challengeではものづくり/サービス/インフラ各部門でコンテストを実施する。
 合わせて展示会を実施する。

2017年12月20,21日
 トライアル大会開催
 コンビニ自動化/接客/トイレ掃除/商品補充

2018 TOKYO 10月17日～21日
 (出場予定)

2020 AICHI/FUKUSHIMA 8月9日

Robotics for Happiness
 人間とロボットが共生し協働する世界の実現へ

Industrial Robotics (ものづくり) | Service Robotics (サービス) | Disaster Robotics (インフラ・災害対応) | Junior (ジュニア)

World Robot Summit

2018 2019 2020 2021

World Robot Summit 2018 TOKYO
 2018年 10月17日 - 10月21日
 東京ビッグサイト 東7Bホール

World Robot Summit 2020 AICHI/FUKUSHIMA
 2020年 8月・10月
 愛知県国際展示場(10月予定) | 福島ロボットテストフィールド(8月予定)

主催 経済産業省 NEDO



World Robot Summit (総称:WRS)は、人間とロボットが共生し協働する世界の実現を念頭に、世界のロボットの知恵を集めて開催する競演会です。

WRSでは、ロボットの競技会「World Robot Challenge」と、最新のロボット技術を示す「World Robot Expo」を介して、世界中のロボット関係者が一堂に集まり、リアルな日々の生活、社会、産業分野でのロボットの社会実装と研究開発を加速させることを目的とします。

競技会
World Robot Challenge (WRC)

展示会
World Robot Expo (WRE)

WRC: ものづくりカテゴリー, サービスカテゴリー, インフラ・災害対応カテゴリー, ジュニアカテゴリー

WRE: 一般企業、大學生、研究機関等が出展、展示を行う一般公開エリア。ロボット導入の事例を世界発信するものとします。競技参加団体、協賛企業などの独自ブースの他に、競が主催する最新のロボット関連展示を行う予定。

カテゴリ	種目	競技内容	実施場所
ものづくり	製品組立チャレンジ	工業製品等の組立に必要な技術要素を含んだモデル製品を早く正確に組立	
サービス	パートナーロボットチャレンジ (家庭内の各種作業実演チャレンジ)	家庭における片付け(整理整頓、収納等)や留守番対応	愛知県国際展示場
	フューチャーコンビニエンスストアチャレンジ (店舗における各種業務の自動化チャレンジ)	食品など複数種類の商品の抽出・入替、接客や従業員とのインタラクション、トイレの清掃	
インフラ・災害対応	フロント災害予防チャレンジ	数種のインフラ設備目に基づく点検メンテナンス(バルブ開閉等)	福島ロボットテストフィールド
	トンネル工事・災害対応-掘削チャレンジ	トンネル工事を実施した掘削収集、集塵対応 (人命救助、災害復旧業務等)	
ジュニア (小学生以下)	災害対応-防災訓練チャレンジ	災害予防・対応に必要な各種操作性評価 (移動能力、センシング能力、情報収集能力、意思決定能力、遠隔操作能力、災害復旧能力、耐久性)	
	スクールロボットチャレンジ	学校現場においてニーズのありそうなタスクとそれを表現するロボットをプログラミング	愛知県国際展示場
ホーム	サービスカテゴリーのパートナーロボットチャレンジ	サービスカテゴリーのパートナーロボットチャレンジと同様のタスクを設定しロボットを製作	
	ホームロボットチャレンジ		

*2020年競技内容については、開催までの調整であり、技術の進歩や2018年の実績結果も踏まえて確定。

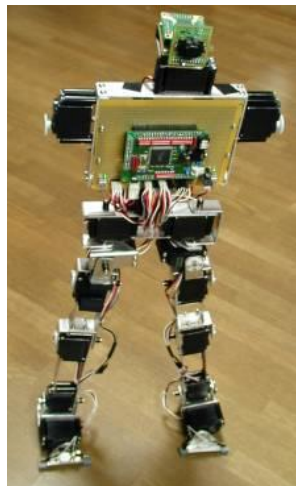
World Robot Summit (WRS) 運営事務局
 開催・ご協賛に関するお問い合わせ
 公式HP <http://worldrobotsummit.org>
 Facebook <https://www.facebook.com/worldrobotsummit/>
 YouTube https://www.youtube.com/watch?v=8bvf_w5Cu2s

ROBO-ONE技術の進化

見た目のロボット技術が中心に進化を遂げた。メカ・サーボに見られる進化は軽量化から実用性へ移行し始めた。

ヒューマノイドプロジェクト開始 2001年1月24日

NR-1が3歩あるく 2001年6月25日

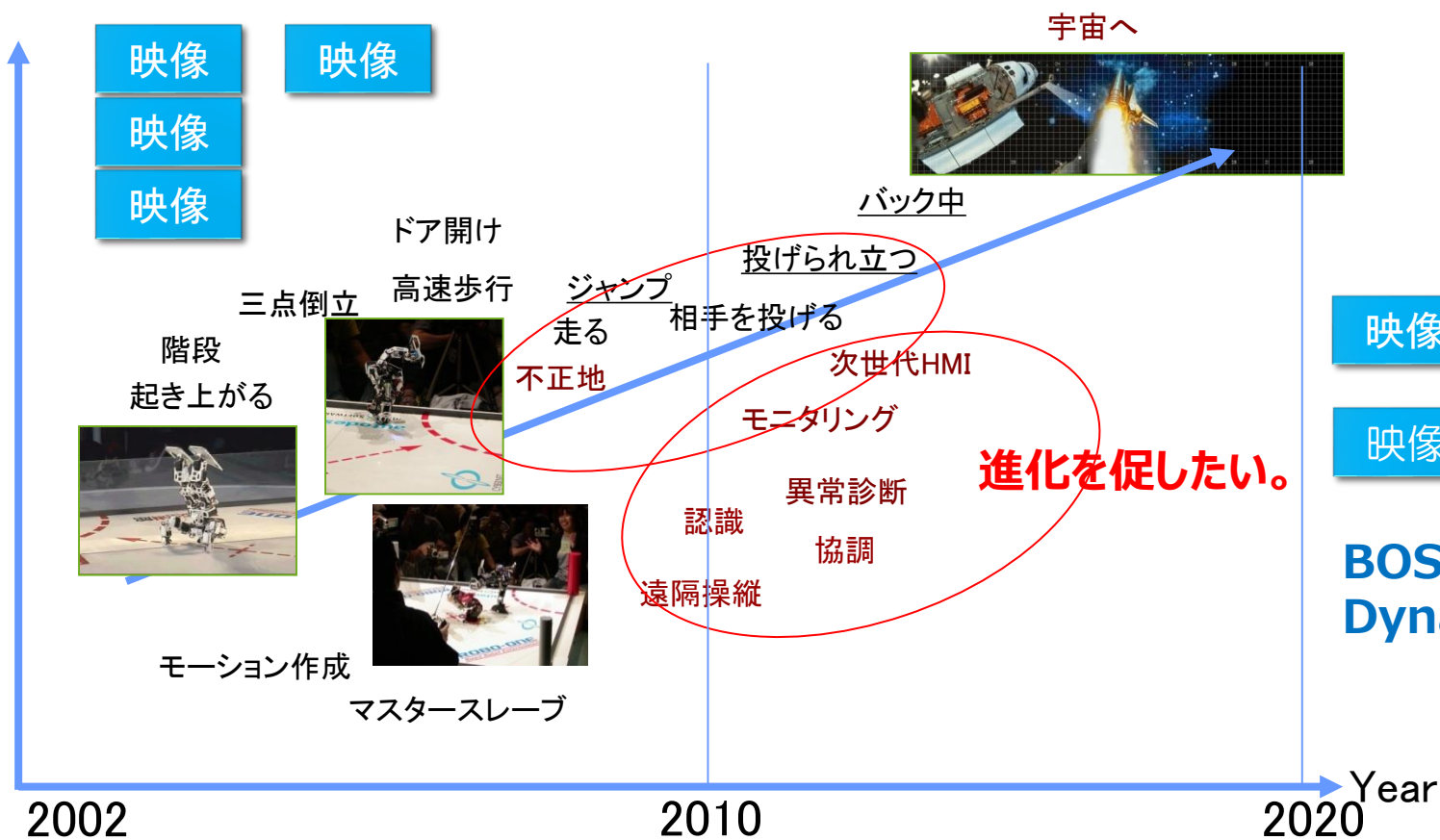


- 映像
- 映像
- 映像
- 映像

ROBO-ONEイベント 2001年9月29日

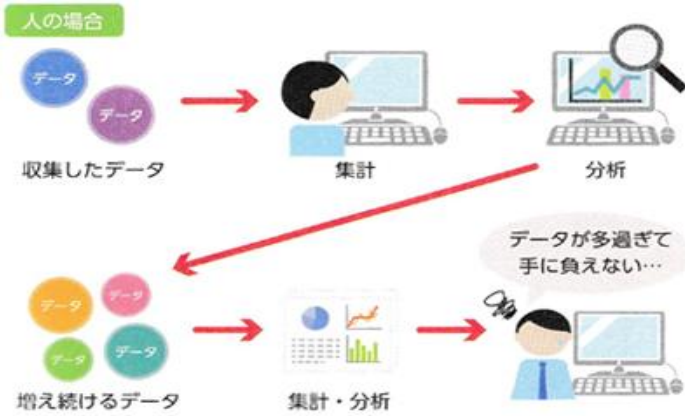
ROBO-ONE第1回大会 2002年2月2日

技術

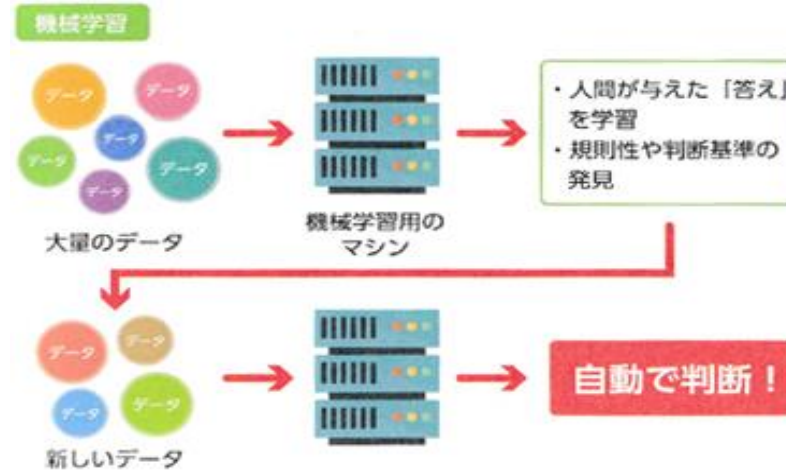


深層学習の登場

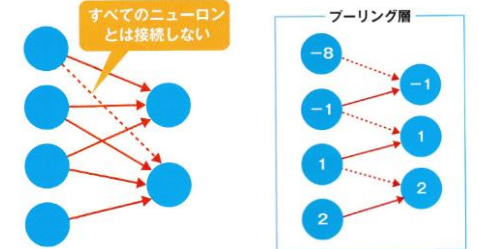
EY総合研究所レポート「人口知能が経営にもたらす創造と破壊」より



ニューラルネットを使った機械学習
深層学習の登場

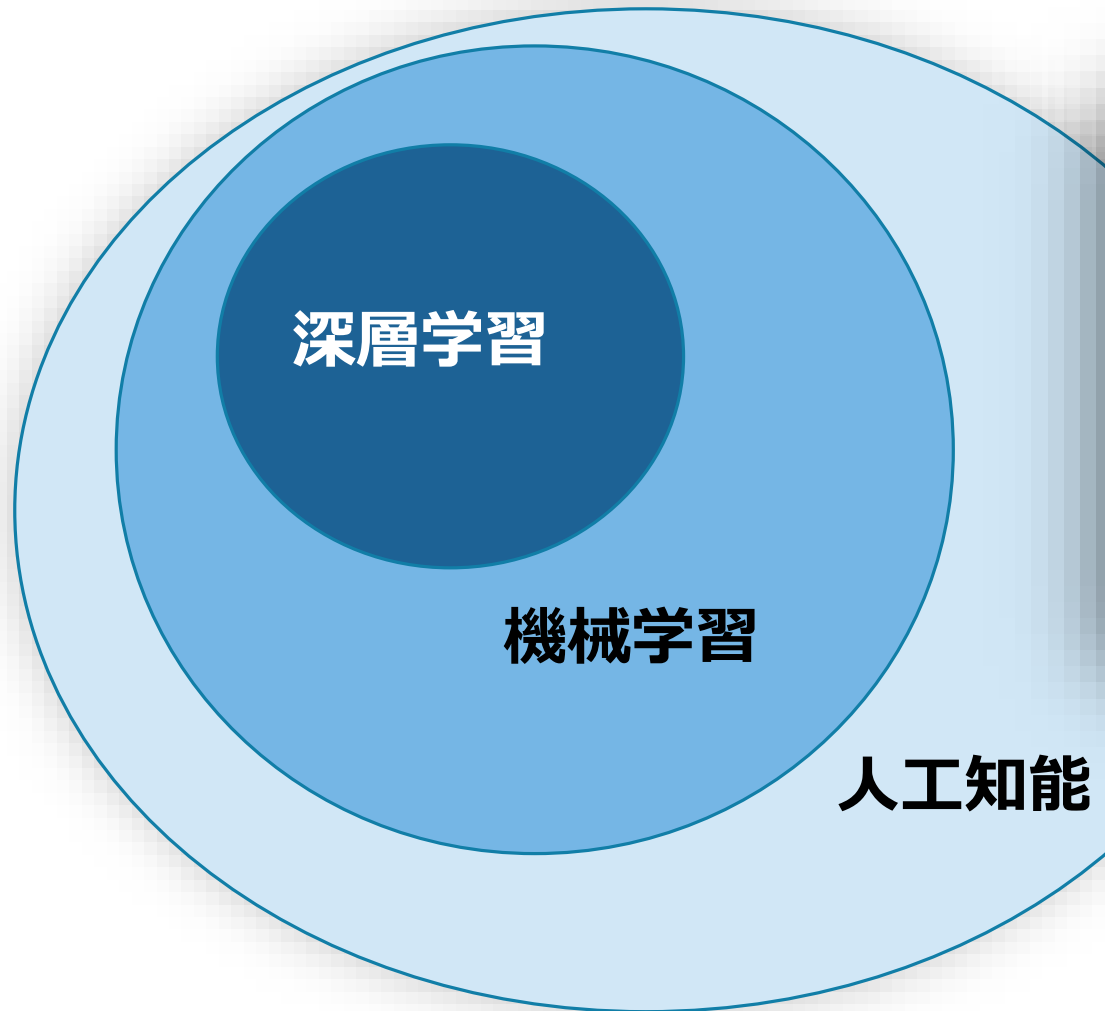


特徴の発見
大量のデータ
ルールに基づいた判断

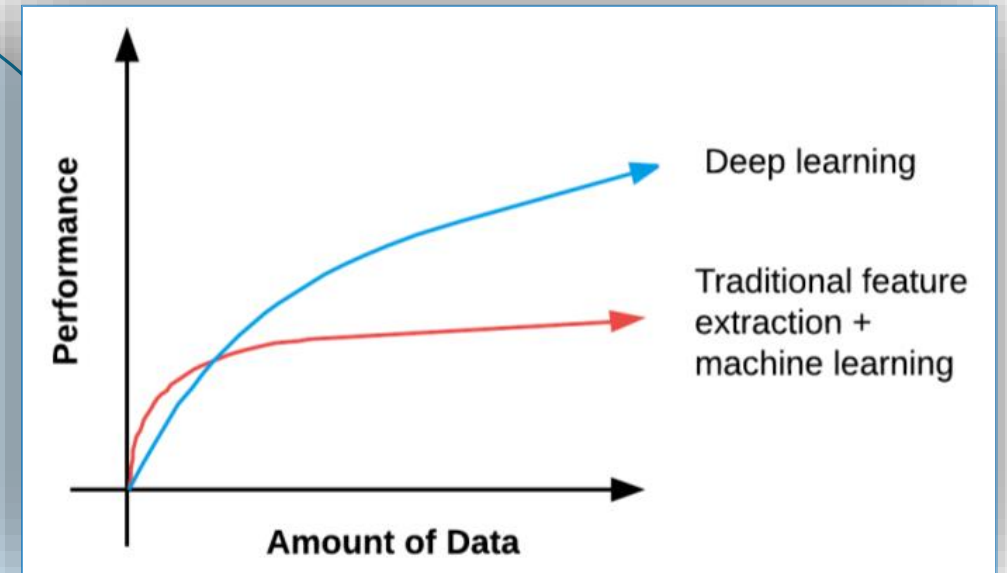


ニューラルネットを使った機械学習イメージ

人工知能と深層学習



機械学習と深層学習



人工知能の進化

ニューラル・ネットワーク

教師あり
 --パーセプトロン
 --バックプロパゲーション
 --アソシアトロン

教師なし
 --自己組織化マップ(SOM)
 --ボルツマン・マシン

データ・マイニング

--If-Then
 --主成分分析
 --クラスタ分析
 --サポート・ベクタ・マシン(SVM)

学習・進化

--Qラーニング
 --山登り法
 --遺伝的アルゴリズム

ディープニューラルネット:ニューラルネットを取り入れた機械学習。画像の分類問題で人間を超えたということで、これをきっかけに第三次人工知能ブームの引き金となった。CNNが進化させた。

CNN(畳み込みNN):画像認識

AE(auto encoder):ノイズ除去や画像生成

RNN(リカレントNN):未来予測や自動作文

ディープ強化学習:ぶつからない車

If-Thenルール:もし〇〇だったら□□をなさというルールがたくさん並んでいるも。

Qラーニング:行動することにある決められた報酬(Q値)が得られるようになっていて、報酬が多く得られるように、動作を更新する方法です。

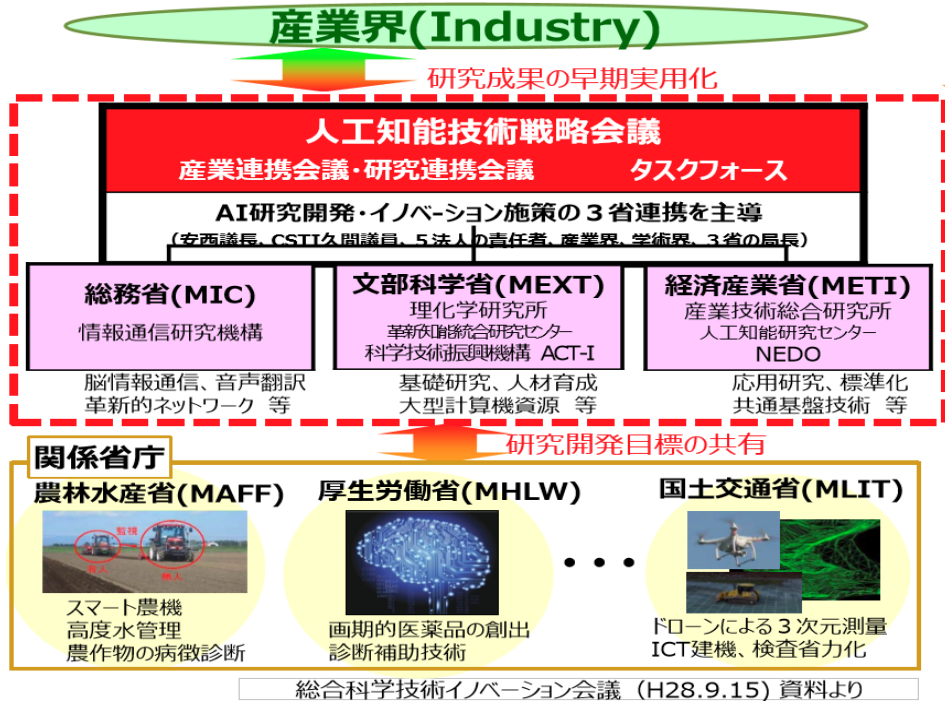
山登り法:ロボットがどのくらい良い動作をしたかを記録しておき、動作パラメータを少しだけ変えて実験します。その結果が良ければ変えた動作パラメータを採用しそうでなければ戻すことを繰り返します。

人工知能技術戦略会議とロードマップ

http://www2.nict.go.jp/ais/ais_sympo.html

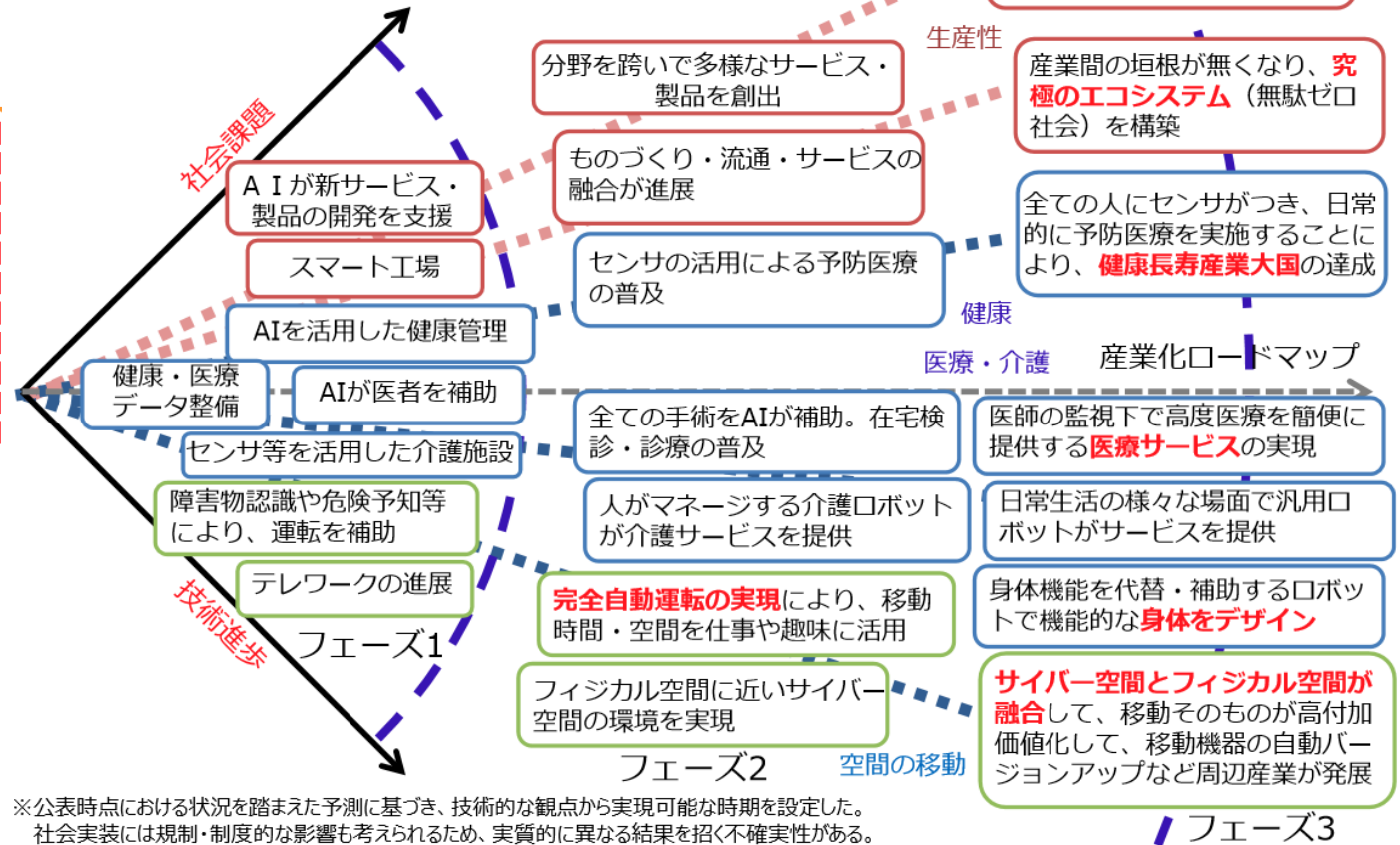
<http://www2.nict.go.jp/ais/anzai.pdf>

人工知能技術戦略会議と関連組織



2. 研究開発目標と産業化ロードマップ

<http://www.nedo.go.jp/content/100862412.pdf>



ARGOS Challenge team

株式会社移動ロボット研究所

ロボット技術により夢を実現し
安全で安心な社会の実現に貢献する

株式会社 移動ロボット研究所
Field and Service Robotics



企業概要

株式会社移動ロボット研究所の企業基本情報、グループ会社、交通アクセス、マップなどのご案内は、こちらから。



業務案内

株式会社移動ロボット研究所は、幅広い分野でロボットの開発を展開しております。ロボットの開発から運用コンサルティングまで、ロボットに関する事はお任せください。



製品紹介・実績紹介

株式会社移動ロボット研究所 製品紹介



映像

映像

映像

映像

東北大学田所研究室 /永谷研究室



Welcome Message
未来科学技術共同研究センターの永谷です。当研究室では、自然環境や災害現場といった実フィールドで活動可能な無人探査用フィールドロボットの研究開発を積極的に進めております。現在、国内外で、自然災害が頻発しており、フィールドロボットの研究開発には、大きな期待が寄せられています。この期待に応え、「役に立つロボット」を実現することが、当研究室の目標です。

研究室を志望する方へ
READ MORE

筑波大学 坪内研究室



ロボ研 トピックス

2017年度 ロボ研春のBBQ
日程: 2017/05/13
場所: ゆかりの森

研究室の大掃除のため、下記の期間はホームページが閲覧できなくなります。
日程: 2017/04/03-04/04

2016年度 第3回 山彦シンポジウム
日程: 2017/02/22-2017/02/24
場所: 国立オリンピック記念青少年総合センター



English

トップ

サイトマップ

概要

構成員

研究テーマ

論文

ムービー

行事

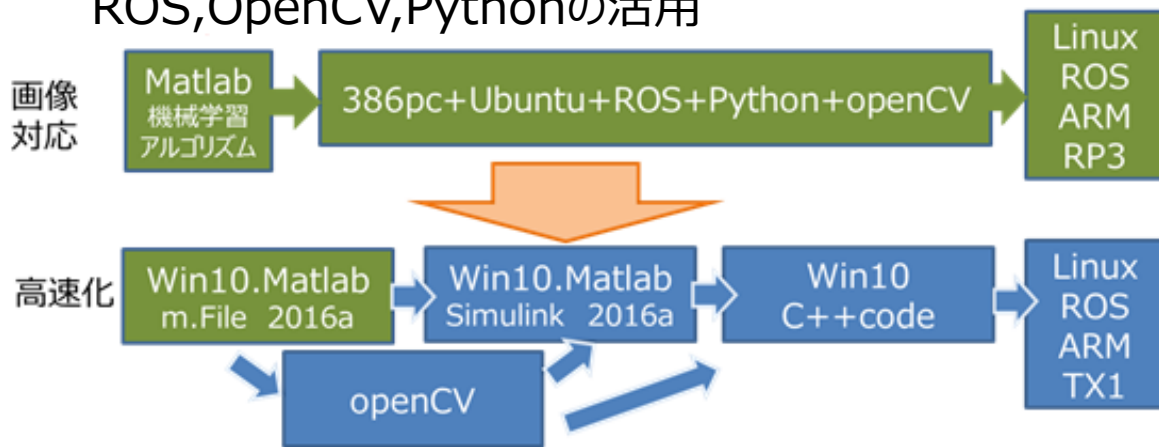
交通アクセス

robots-2

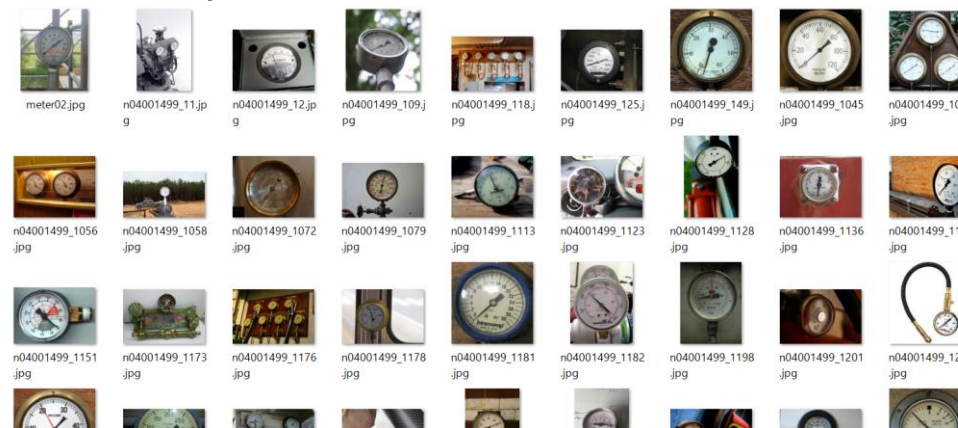
ARGOS Challenge

完全自律型移動ロボットによる石油プラント
監視ロボットコンテストに参加。

ROS, OpenCV, Pythonの活用

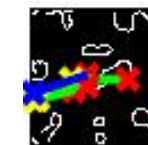
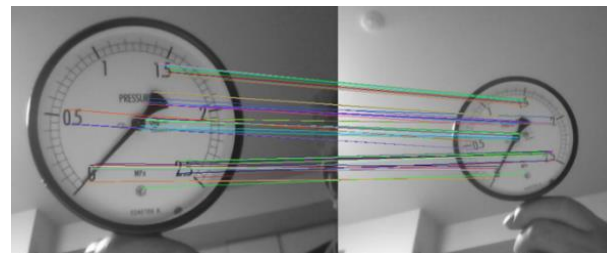
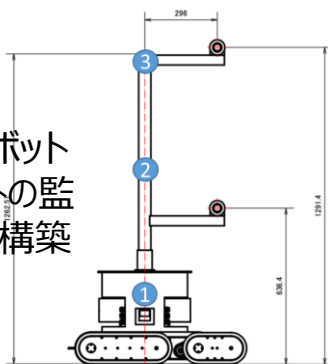


機械学習/深層学習





画像処理/深層学習がAirRCのキーテクノロジーとなる。

完全自律ロボット
によるプラントの監視システムを構築



ARGOS

ARGOSチャレンジにおいて、画像処理/深層学習を使ってメーターの針を読んだり、バルブの位置を読むプログラム開発を行いました。

Designation	Starting Time (HH:MM:SS:FFF)	Ending Time (HH:MM:SS:FFF)	Duration (MM:SS:FFF)	Position (X,Y,Z)	Value	Status	File	Description
Action:check point status assessment CP16 (algorithm 1/2 accepted)	16:54:40:529	16:54:55:979	00:00:15:449	-6.288, 3.423, 0.040	0.17			 0,1
Action:check point status assessment CP16 (algorithm 2/2 rejected)	16:54:59:529	16:55:00:598	00:00:01:068	-6.288, 3.423, 0.040	0.481783843 209 bar	Al		 offset x=-0.02m y=-0.02m confidence=0.567668140466
Detection: status alarm	16:56:01:776			-6.311, 3.343, 0.040		Emergency stop		

Deep Learning

Key Point Matching

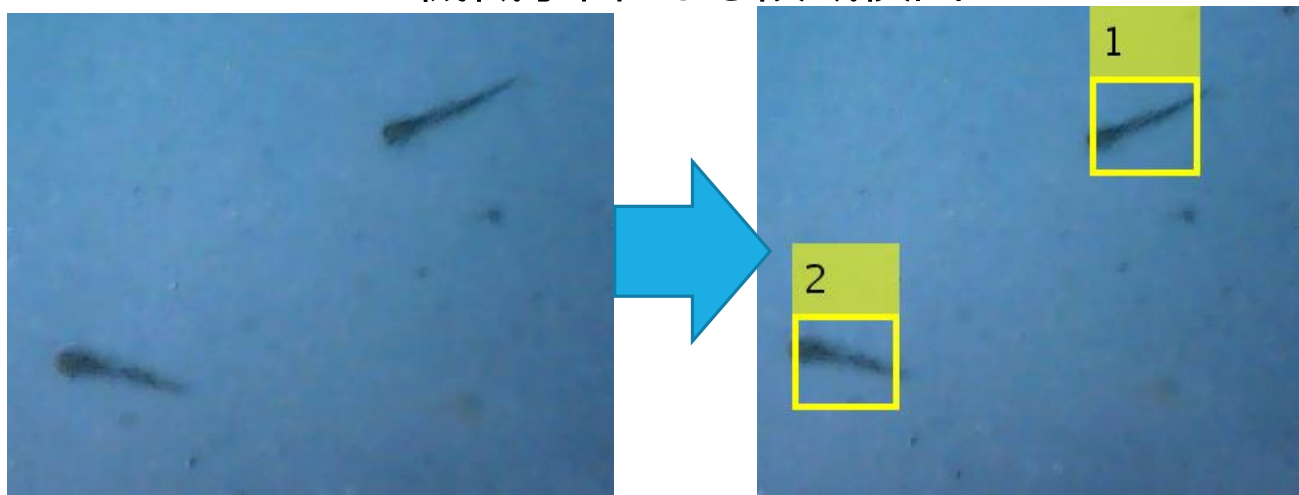
らんちゅう選別ロボット



らんちゅうの選別システムに画像の深層学習活用の基本がある。



機械学習による領域検出



深層学習による善し悪しの判別



吸い取りメカニズム



静止状態の判定から動的判定の研究へ移行

感情の入らない効率的な選別ができる。

トマトのモデル化



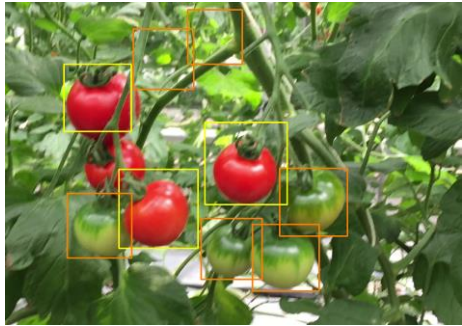
株式会社プラントライフシステムズ ⇔ 技術顧問

実演

栽培支援ソフト + 最少のセンサー + 特殊水耕システムで
安価で非常に高い糖度のトマトができるシステムの確立

AiRRCが担当

**トマト無収穫時期
病気などの判定へ
深層学習適用推進中**



ハウスでの生育



最小限のセンサ



糖度
光合成
水分
葉上CO2



特殊水耕システム

等価状態で
シミュレーション

MBD : モデルベース開発を適用

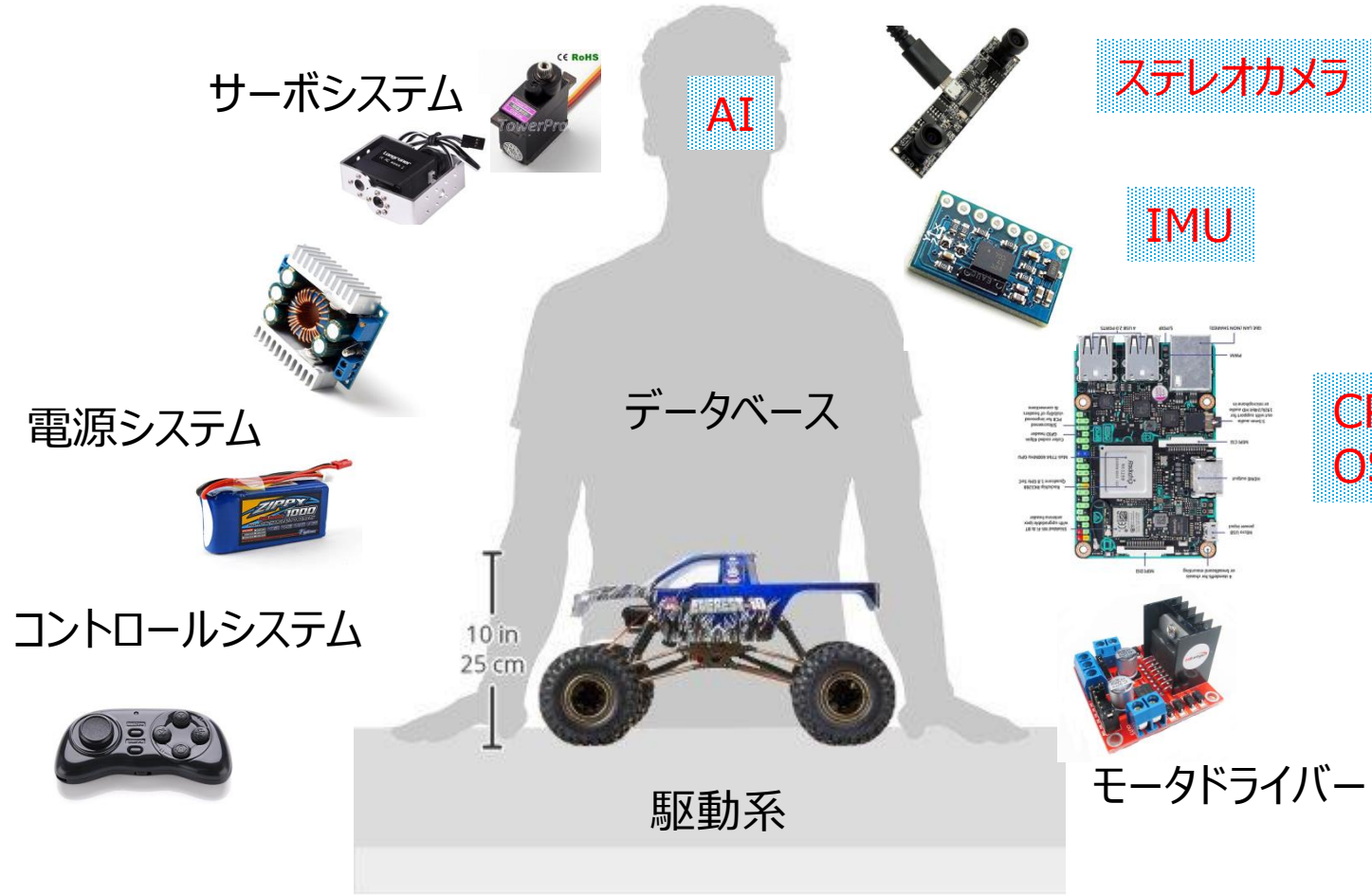
生育アルゴリズム
+ 生育支援

自動車開発等での制御での主流の**プログラム
(モデル予測&ディープラーニング)**を活用



実際の生体状態を把握コンピューター上で表現
非常に高い精度で高品質な作物ができる。

ロボットシステムの要素技術



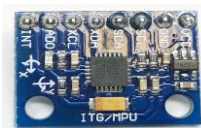
高度な部品の低価格化が進んでおり、高度なロボット製作の可能性が出てきた。

IMU

低価格IMUの調査結果、BNO-055が優れていることが分かった。



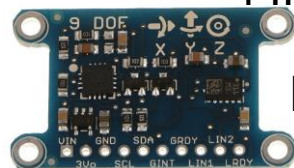
MPU9250 BMP280



MPU-6050



MPU-9250



Lovoski IMU



CMPS11-Tilt補償磁気コンパス



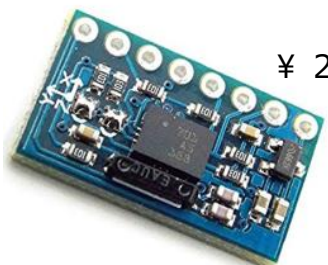
10自由度 IMU

FOG (Fiber Optical Gyro)



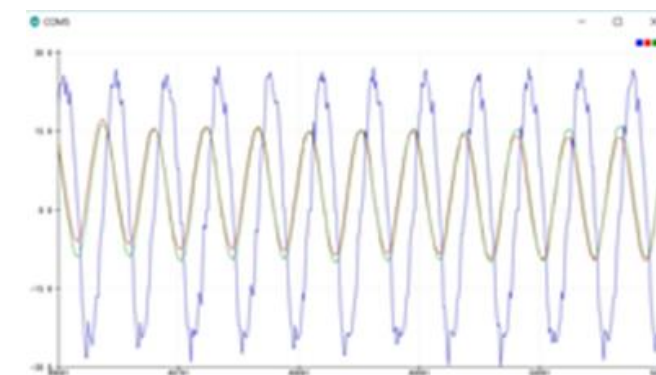
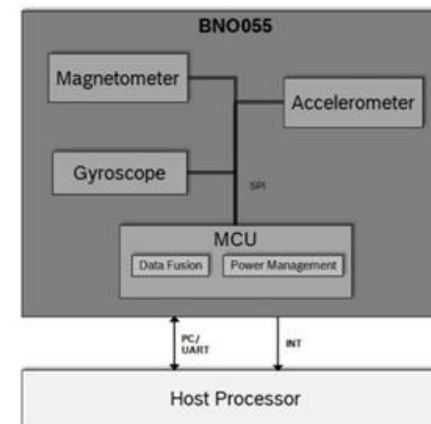
GY-BNO-055: **BNO055** 3軸加速度センサー + 3軸ジャイロセンサー + 3軸地磁気センサーモジュール

BNO055



¥ 2,450

Operating Mode	Available sensor signals			Fusion Data		
	Accel	Mag	Gyro	Relative orientation	Absolute orientation	
Non-fusion modes	CONFIGMODE	-	-	-	-	
	ACCONLY	X	-	-	-	
	MAGONLY	-	X	-	-	
	GYROONLY	-	-	X	-	
	ACCMAG	X	X	-	-	
	ACCGYRO	X	-	X	-	
Fusion modes	MAGGYRO	-	X	X	-	
	AMG	X	X	X	-	
	IMU	X	-	X	X	
	COMPASS	X	X	-	-	X
	M4G	X	X	-	X	-
	NDOF_FMC_OFF	X	X	X	-	X
NDOF	X	X	X	-	X	



オイラー角とジャイロ積分角との比較

ロボットと人工知能の連携

クラウドでロボットやセンサーが繋がり、より高度なサービスを提供する「クラウドロボティクス」

ArduinoでROBO-ONEのロボットは動く
次は知能を追加したい。

自律神経を持つロボット



知能:自分で考えるかクラウドを使うか?

情報:数値化するか生情報か?

ROSなどのミドルウェアによる体内LAN

ロボットだから得られる情報がある。

Mother

人工知能:学習による進化
他の人工知能と連携



google home



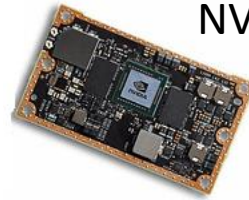
Aibo



画像処理CPUについて



RasPi 3
¥5,000.-



NVIDA TX2



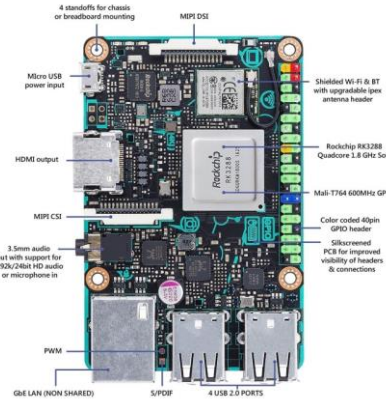
ベアボーンPC i7

¥39,000.-

NUC i7*4
¥190,000.-



RasPi zero
¥3,000.-



ASUS TinkerBoard
¥9,000.-



Intel Z8350
¥ 25,724



Ubuntu16.04LTS+ROS(Kame)/Stereo Camera対応コストからRasPi互換のASUS TinkerBoardにて推進。

画像処理のためのOS

ロボットに適用を前提とすれば、ROSということになる。

現在はUbuntu16.04+kinetic

まずはPythonからスタートするとすれば、

Python2.7+OpenCV3.4

1.Ubuntu16.04 install

Tinker:Armbian install

<http://gihyo.jp/admin/serial/01/ubuntu-recipe/0488>

Nuc / up :from USB

<https://kledgeb.blogspot.jp/2016/04/ubuntu-1604-1-ubuntu-1604ubuntuubuntu.html>

ROS-kinetic+Python+OpenCV

ROS-kinetic install

<http://wiki.ros.org/kinetic/Installation/Ubuntu>

レポジトリ追加

```
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" >
/etc/apt/sources.list.d/ros-latest.list'
```

認証キー追加

```
sudo apt-key adv --keyserver hkp://ha.pool.sks-keyservers.net:80 --recv-key
0xB01FA116
```

```
sudo apt-get update
```

フルインストール

```
sudo apt-get install ros-kinetic-desktop-full
```

イニシャライズ

```
sudo rosdep init
```

```
rosdep update
```


Pythonを使ってみる

Anacondをインストール

<https://www.continuum.io/>からダウンロード。

Windowsでも使えます。

Anaconda 4.3.1

For Windows

Anaconda is BSD licensed which gives you permission to use Anaconda commercially and for redistribution.

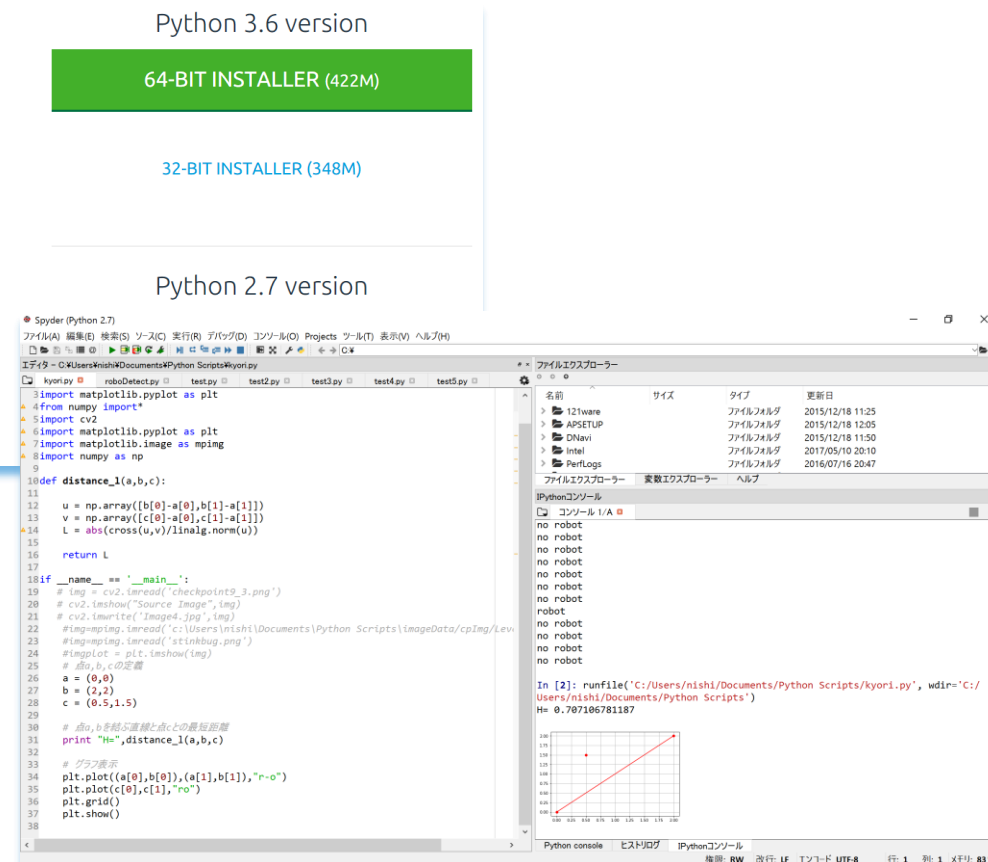
Changelog

1. Download the installer
2. Optional: Verify data integrity with [MDS](#) or [SHA-256](#) [More info](#)
3. Double-click the `.exe` file to install Anaconda and follow the instructions on the screen

Behind a firewall? Use these [zipped Windows installers](#)

Spyderを使うとMATLABな感じ。
NumPyとmatplotlibなどの
ライブラリーが有難い。

動画を読み込めない場合は
`Pip install opencv_python-2.4.13.2-cp27-cp27m-win_amd64.whl`



ライブラリーのいろいろ

Python 標準ライブラリ

<https://docs.python.jp/2.7/library/index.html>

<http://programming-study.com/technology/python-library/>

NumPy

NumPyは科学技術計算などで配列や行列の演算を高速で行うために使われます。研究などで数学的な計算をしたいときに、非常に有効なライブラリとなっています。

```
import numpy
n_array = numpy.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
print(n_array)
```

matplotlib

matplotlibを使えばPythonでデータをグラフにプロットできるようになります。
グラフの表示にはmatplotlib.pyplotモジュールのplot()およびshow()を使用します。

```
from matplotlib import pyplot
pyplot.plot(x軸, y軸)
pyplot.show()
```

Program記述方法

chmod u+xで実行権を与える

漢字も使える

numpyをimport

matplotlib.pyplotをimport

defで関数を宣言

インデントによってブロック化

Mainからスタート

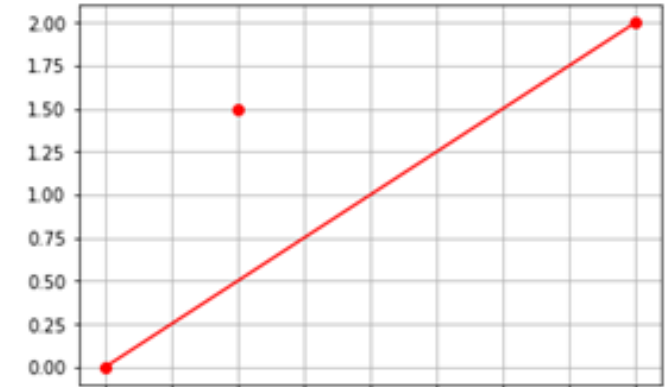
配列をセット

matplotlib関連

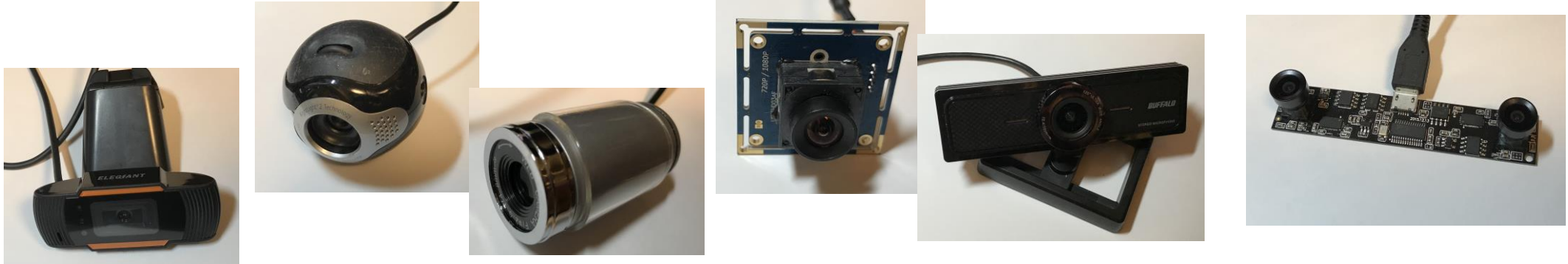
```

1#!/usr/bin/env python
2# -*- coding: utf-8 -*-
3
4import numpy as np
5import matplotlib.pyplot as plt
6
7def distance_1(a,b,c):
8    u = np.array([b[0]-a[0],b[1]-a[1]])
9    v = np.array([c[0]-a[0],c[1]-a[1]])
10   L = abs(cross(u,v)/linalg.norm(u))
11   return L
12
13if __name__ == '__main__':
14   # 点a,b,cの定義
15   a = (0,0)
16   b = (2,2)
17   c = (0.5,1.5)
18   # 点a,bを結ぶ直線と点cとの最短距離
19   print "H=",distance_1(a,b,c)
20
21   # グラフ表示
22   plt.plot((a[0],b[0]),(a[1],b[1]),"r-o")
23   plt.plot(c[0],c[1],"ro")
24   plt.grid()
25   plt.show()

```



カメラの選定



No	走行用	ステレオ	QR/AR	名称	視野角	解像度at30fps	自動露光	自動焦点	その他
1	×	×	×	ELP HDの2megapixel1080P	20~30	640×480/320×240	ややラフ	なし	ARGOSで使用
2	×	×	◎	ELP-USBFHD01M-L36	30	640×480/320×240	ややラフ	なし	ARGOSで使用
3	○	○	×	ELP 720P デュアルレンズ	100	640×480/320×240	暗いところに弱い	なし	レンズ歪み大
4	◎	◎		BUFFALO 200万画素WEBカメラ	120	640×480/320×240/160×120	良い	yuy2以外は遅れがある	yuy2にて使用のこと
5	×	×	×	XQB	30	640×480/320×240/160×120	良い	なし	
6	×	×	×	WEBカメラ ELEGANT U	20	640×480/320×240/160×120	良い	良い	
7	×	×	×	logcool	45	640×480/320×240/160×120	良い	良い	
8	×	×	×	PS2cam	100	640×480/320×240/160×120	良い	良い	75fpsまで可/切り替えzoom/入手困難
9	×	×	×	Cimkiz USB 2.0 HDウェブカメラ	30	640×480	ややラフ	なし	
10	◎	◎	×	ELP 0.3MP YUY2 MJPEG	100	640×480/320×240	良い	良い	レンズ湾曲
11	◎	◎	×	ELP Full Hd ウェブカメラ	100	640×480/320×240	ややラフ	良い	
12	×	×	×	XCSOURCE マイクスタ	45	640×480/320×240	良い	なし	
13	×	×	×	ELP 5メガピクセル工業U	60	2592x1944高HD解像度	良い	良い	耐熱性なし

画像処理のテクニック

```
SaveMovie.py (~/.catkin_ws/src/roc/scripts) - gedit
開く(O) 保存(S)

import numpy as np
import cv2
import time
cap = cv2.VideoCapture(0)
cap.set(cv2.CAP_PROP_FPS, 30)
cap.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)
fourcc = cv2.VideoWriter_fourcc(*'MJPG')
video = cv2.VideoWriter('video.avi', fourcc, 30, (40, 480))
while(cap.isOpened()):
    ret, frame = cap.read()
    if ret==True:
        #*/+-
        eframe=frame*2-50
        print eframe.max(),int(eframe.mean()),eframe.min()
        #resise
        rframe=cv2.resize(frame,(64,48))
        #trim
        tframe=frame[0:480,640/2-20:640/2+20]

        video.write(tframe)
        cv2.imshow('frame1',rframe)
        cv2.imshow('frame2',tframe)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break
    else:
        break
# Release everything if job is finished
cap.release()
video.release()
cv2.destroyAllWindows()

Python  タブ幅: 8  (5行、30列)  [挿入]
```

実演

画像の取り込み保存
モルフォロジー
エッジ処理
マッチング
色判断
動きの判断
など

機械学習

実演

OpenCVで物体検出器を作成する

https://www.pro-s.co.jp/engineerblog/opencv/post_6202.html

学習

- 物体を検出する際、あらかじめ画像から、特徴量を抽出する。

推論/検証/検出

- 学習済の特徴に合う部分がないかを画像上を探索する。

カスケード分類器

haartraining (ハールトレーニング)

Haar-Like特徴による機械学習

traincascade (トレインカスケード)

haartrainingのあとに作られたもの

Haar-Like特徴・LBP特徴・**HOG特徴**から1つを選択

輝度の勾配方向の分布

- 正解画像は7000枚以上、不正解画像は3000枚以上

Deep Learning

実演

- 1.機械学習 Hog⇒Haar
- 2.Chainer 浅層学習/搭載型
- 3.深層学習 多くの画像深層学習クラウドができた。
Labellio(京セラ) Zinrai(Fujitsu)
Custom Vision Service(Microsoft)
Watson(IBM) etc.



Labellio(画像の深層学習クラウド)昨年暮れ TensorFlowで再スタート。大幅に精度向上。

トマトの病気の深層学習

<https://www.labell.io/datasets/>

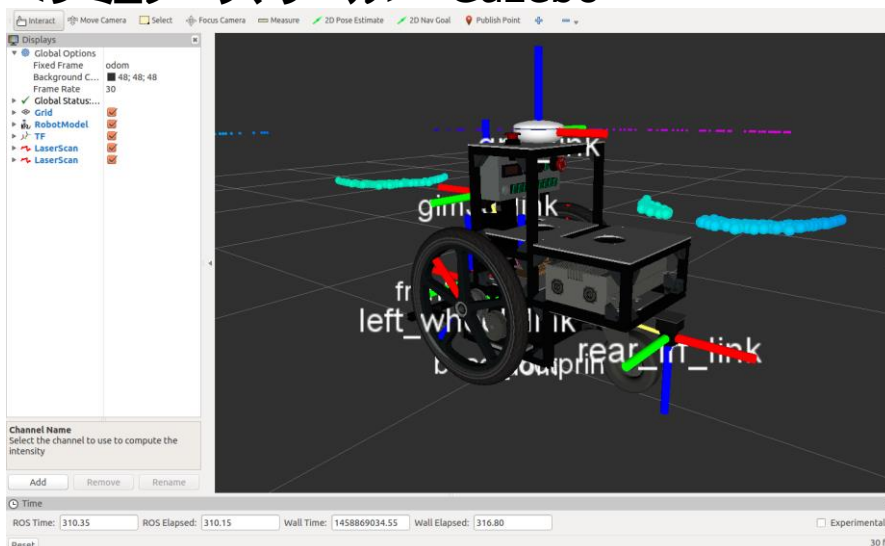
ROS (Robot Operating System)

ロボットの統合的モデルベース開発ツール
Ubuntu16.04LTSにROS(Kame)を搭載。

<ROSとは>

1. プロセス間通信のための通信ライブラリ
2. プログラムをコンパイルするためのビルドシステム
3. リアルタイム処理ではない。ROS2でリアルタイムへ

<シミュレータ、ツール> Gazebo



<ライブラリー>

tf: 座標変換、フレーム作成
pcl : ポイントクラウドライブラリ
OpenCV:画像認識ライブラリ
usb_cam:USBカメラとの接続
などのライブラリが充実しています。

Point cloud



ステレオカメラの校正とPoint Cloud

<Install>

```
http://bril-tech.blogspot.jp/2016/11/ros9-ros.html
```

```
sudo apt-get install -y ros-kinetic-udev-camera
```

```
sudo apt-get install -y ros-kinetic-image-*
```

<Calibration>

```
roslaunch uvc_camera uvc_stereo_node _left/device:=/dev/video0 _right/device:=/dev/video1 _fps:=30  
_skip_frames:=2 _width:=320 _height:=240
```

<matlab check>

```
roslaunch camera_calibration cameracalibrator.py --size 9x6 --square 0.03 right:=/right/image_raw  
left:=/left/image_raw left_camera:=/left right_camera:=/right
```

<opencv:http://opencv.jp/sample/pics/chesspattern_7x10.pdf>

<Point Cloud>

```
ROS_NAMESPACE=stereo roslaunch uvc_camera uvc_stereo_node _left/device:=/dev/video0  
_right/device:=/dev/video1 _fps:=30 _skip_frames:=5 _width:=320 _height:=240
```

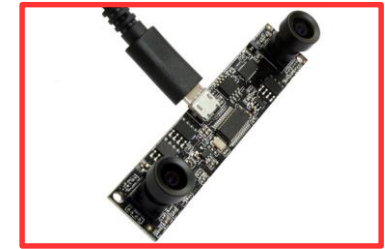
```
ROS_NAMESPACE=stereo roslaunch stereo_image_proc stereo_image_proc
```

```
roslaunch image_view stereo_view stereo:=/stereo image:=image_rect_color _approximate_sync:=True
```

<reconfigure>

```
roslaunch rqt_reconfigure rqt_reconfigure
```

実演



Realsense

実演



<How to install>

<https://qiita.com/alfredplpl/items/0b1d9d1d369032512d1e>

インテル®RealSense™デバイス用ROSラッパー2.0 (ビルド2.0.3)

https://translate.googleusercontent.com/translate_c?depth=1&hl=ja&prev=search&rurl=translate.google.co.jp&sl=en&sp=nmt4&u=https://github.com/IntelRealSense/librealsense/tree/development/wrappers/python&xid=17259,15700022,15700124,15700149,15700168,15700173,15700186,15700191,15700201,15700205&usg=ALkJrhhnvspfAA_a_AkK6CVkNG3QdNzFew

PCL(point cloud library)

PCL(Point Cloud Library)+ROSで3次元画像処理入門
<https://karaage.hatenadiary.jp/entry/2017/09/11/073000>

実演

Python PCL (Point Cloud Library)のインストールとサンプル実行
<http://zuqqhi2.com/python-pcl>

ROSで始めるロボティクス(8) — ロボットのナビゲーションを行う
<http://bril-tech.blogspot.jp/2016/10/ros8.html>

<https://pypi.python.org/pypi/PyPCL>

まとめ

画像には多くの情報があり、処理には時間がかかるとのイメージが多かったが、低コスト化、小型化、高速化が進み、ROBO-ONEロボットへの搭載の可能性が出てきた。

まずはこれらをベースに使える処理から、ROBO-剣、ROBO-ONE autoに活用していただけることを願う。

(参考)活用方法は今後、(株)人工知能ロボット研究所サイトにて公開していきます。