

ROS BURGER

ロボット設計製作論 3

18C1034 桑野 雅久

18C1107 堀端 脩平

発表の流れ



概要・コンセプト


技術・ソースコード解説

実演

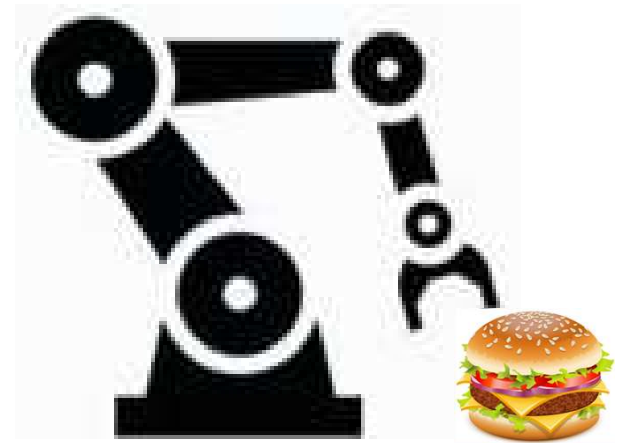
結果・考察

概要

目標は**crane_x7**にハンバーガーの盛り付けを担当させること。



ロボットに
ハンバーガーを
作ってもらおう！



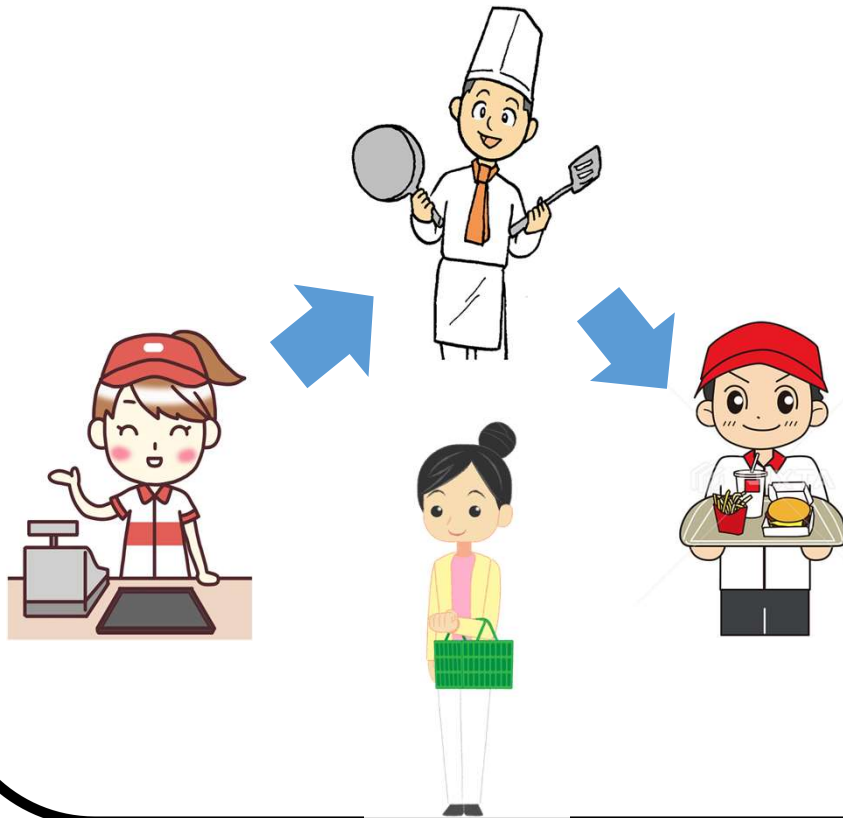
コンセプト

ロボットによるハンバーガーショップの無人化を目指す

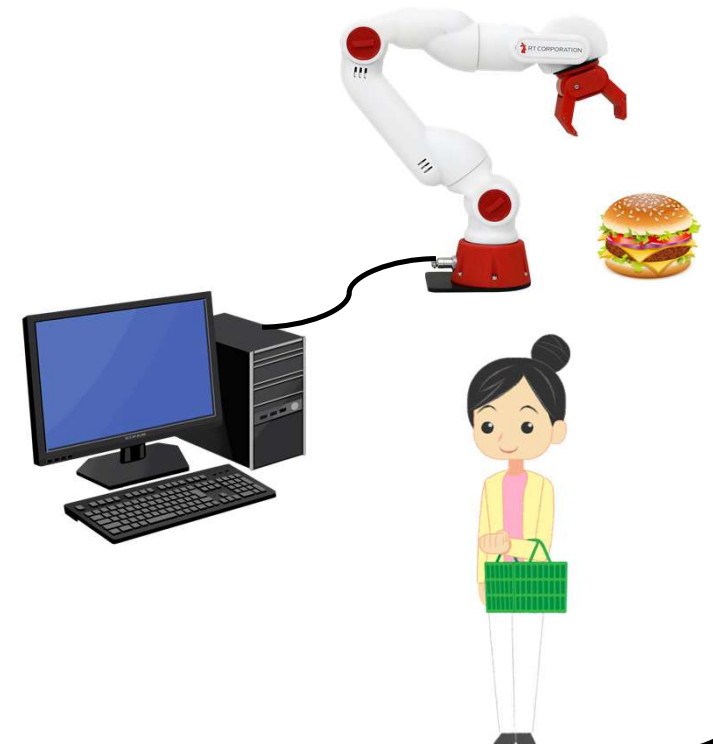


コンセプト

従来のシステム



目指すシステム

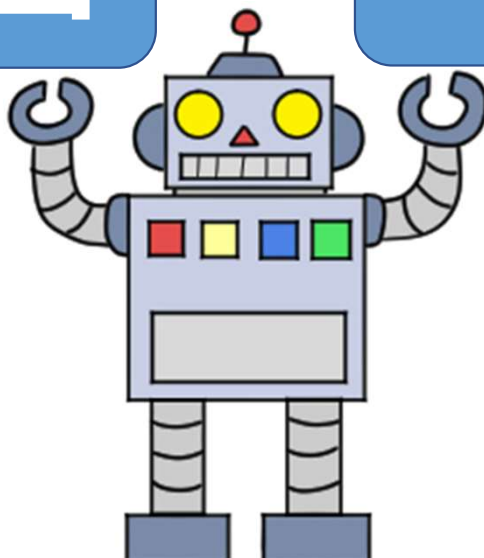


コンセプト

ロボットによる接客を実現させる技術

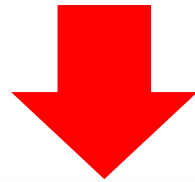
画像検出

GUI



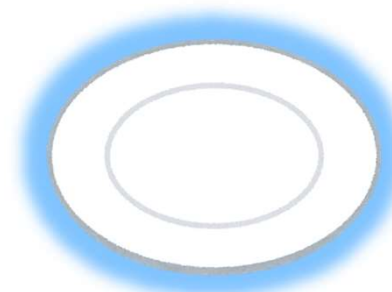
画像検出

皿が準備できていないのに作り始めないでほしい...



皿が置かれたらハンバーガーの作成をスタート

認識にはRealSenseを使用
検出にはDarknet_rosを利用

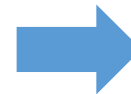


ソースコード解説 Vision編

```
4 def DarknetBboxCallback(darknet_bboxes):
5     print('detection')
6     bbox = darknet_bboxes.bounding_boxes[0]
7     bbox_name = bbox.Class
8     if bbox_name == 'paperplate':
9         print(bbox.Class)
10        app = QApplication(sys.argv)
11        ex = Example()
12        ex.show()
13        sys.exit(app.exec_())
```

受け取った画像の検出データが'paperplate'ならば出力する

```
rospy.init_node("crane_x7_make_hamburger")
rospy.Subscriber('/darknet_ros/bounding_boxes', BoundingBoxes, DarknetBboxCallback)
rospy.spin()
```



検出データを受け取る

GUI

お客様の注文を受けるためには？

お客様の注文をロボットに伝える
インターフェースが必要



PyQtでGUIを作成して、お客様に入力してもらう



ソースコード解説 GUI編

class Example(QWidget):

- 1 目目のウィンドウを作成し表示する.
- ボタンが 4 つある.
 - "Hamburger"をクリック→class Hamburger...に移る.
 - "Cheeseburger"をクリック→class Cheeseburger...に移る.
 - "Specialburger"をクリック→class Specialburer...に移る.
 - "CREATE"をクリック→class CREATE...に移る.



ソースコード解説 GUI編

```
class Hamburger(QMainWindow):
```

2つ目のウィンドウを作成し表示する.

"OK"をクリック→class Food_Stock...の中にある関数を呼ぶ.

```
class Cheeseburger(QMainWindow):
```

Class Hamburger...とほぼ同じ役割

```
class Specialburger(QMainWindow):
```

class Hamburger...とほぼ同じ役割



ソースコード解説 GUI編

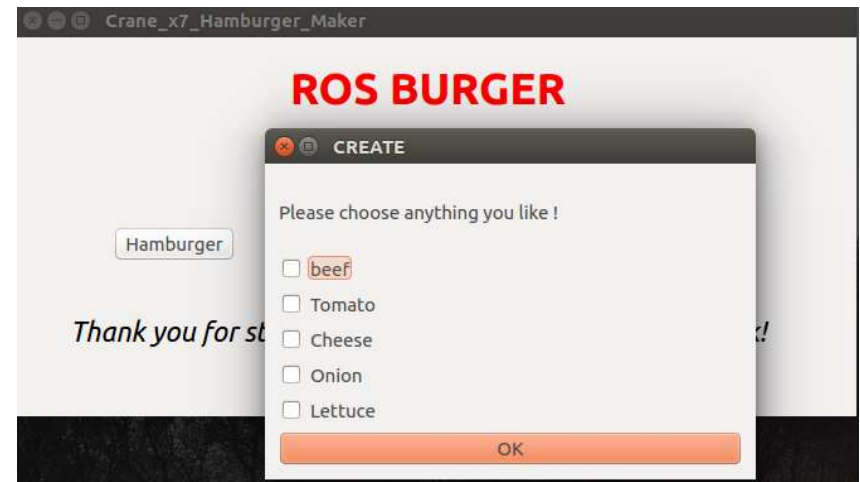
class CREATE(QMainWindow):

- 2つ目のウィンドウを作成し表示する.
- チェックボックスにチェックされた具材だけ実行される.

例) beefにチェックを入れた場合

class Food_Stock...の中に定義されている

def beefだけが呼ばれる



ソースコード解説

motion編

```
class Food_Stock():
```

- ・ Moveitでロボットを制御する.
- ・ 具材ごとに関数で定義されている.
- ・ 関数が呼ばれるとcrane_x7が具材をとり盛り付ける.

例) def beef(self, order):が呼ばた場合

1. 在庫があるかを確認.
2. 在庫があれば具材を取りに行く. なければエラーを表示.
3. プレートに盛り付けをする.

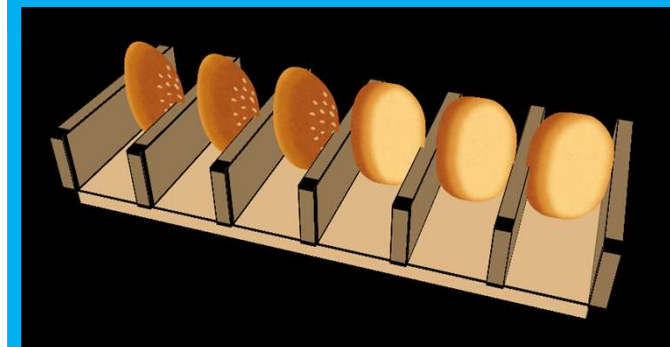
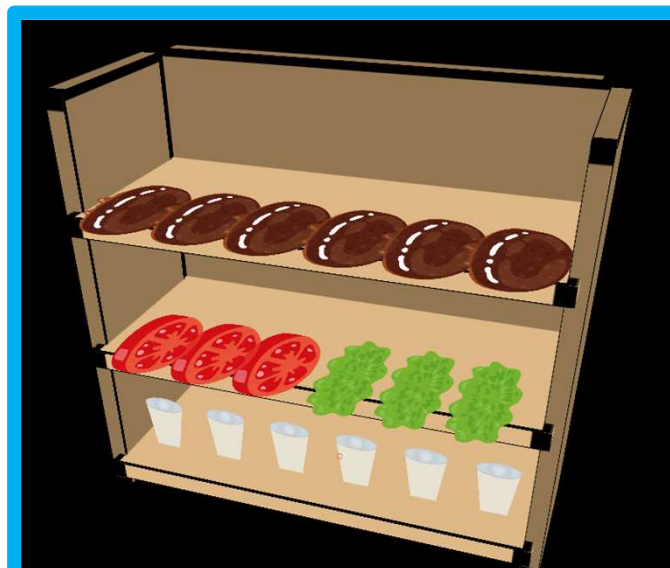
方法

バンズ・具材を置く、右図のようなものを用意する

お客様から注文を受ける

crane_x7のアームで食材を掴む

1つずつ皿に運び、積み上げてハンバーガーを作る

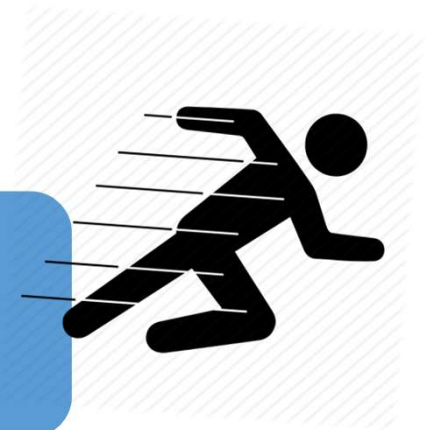


何が面白いのか

技術的に...

アームの軌道や食材の掴み方、持ち上げ方を細かく調整した

他の食材への干渉を無くし、スピーディーにハンバーガーを提供することが出来た



画像検出・GUIを活用

従業員がいなくても、ロボットが接客するのを可能にした



何が面白いのか

社会的に...

ロボットが食材を運び、
綺麗に積み上げることが
できた

ロボットの精度をアピール



お皿が置かれていない
ときはハンバーガーの
作成を始めない

様々な状況に対応し、不
慮の事故を防ぐことがで
きることをアピール



何が面白いのか

余興的に...

人間らしさ

紙コップに入った食材を盛り付ける際、
上下に振る動作を取り入れた。



おもしろさ

ロボットがハンバーガーを作る。
→エンターテインメント



実演



実演動画

(作ったものはスタッフがおいしくいただきました。)

利点と欠点

メリット

人件費・コストの解消

働き手が少なくても
サービスを提供できる
→人手不足の解消

デメリット

初期費用が高い

人間がやったほうが効率
が良い

現段階ではまだトラブルが
多い

結果・考察

- ・食材を掴み運ぶ際、アームから滑ったり、棚との摩擦の影響を受けたりして、少しでも位置がずれてしまうと綺麗に積み上げることが出来なかった。
- ・アームで食材を掴むのに限界がある。
レタスなどの食材は形が不揃いなので、掴むのが難しい。
- ・画像検出は高い精度で認識することができた。
- ・GUIを用いて、ロボットによる接客の無人化を実現できた。

リンク

使用したプログラム(branch : sample1)

https://github.com/GakuKuwano/crane_x7_ros/blob/sample1/crane_x7_examples/scripts/hamburger_maker.py

使い方

```
User:~/catkin_ws[]$roslaunch crane_x7_bringup demo.launch fake_execution:=false
```

```
User:~/catkin_ws/[]$roslaunch realsense2_camera rs_camera.launch
```

```
User:~/catkin_ws[]$roslaunch darknet_ros darknet_ros.launch
```

```
User:~/catkin_ws/src/crane_x7_ros/crane_x7_examples/scripts[]$  
roslaunch crane_x7_examples hamburger_maker.py
```