



WHILL

WHILL Model CR meets ROS - CES2019の舞台裏 -

WHILL株式会社 清水星矢・杉浦光

Who's WHILL?



WHILL + ROS = ?



WHILL meets ROS at CES 2019!



<https://www.youtube.com/watch?v=JVf-OTqWyKM>

Today's topics



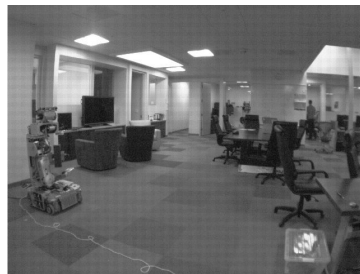
“Detect obstacles in real-time”
【`gpu_stereo_image_proc`】



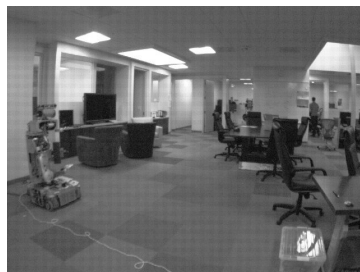
“Sync robot states with
AWS IoT Device Shadow”
【`aws_iot_bridge`】



stereo_image_proc revisited



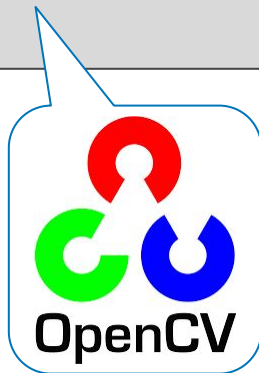
左目画像



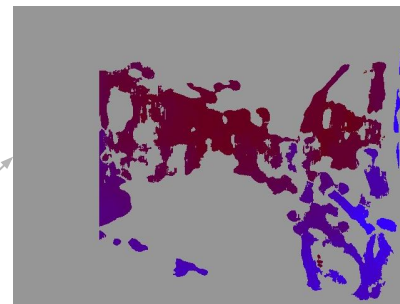
右目画像

stereo_image_proc

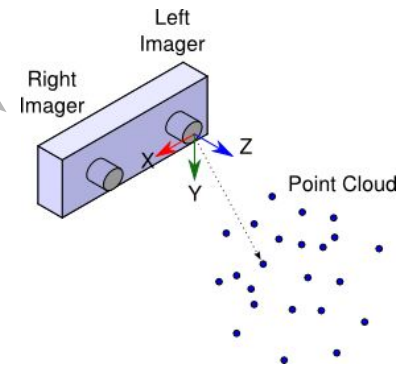
- Block Matching
- Semi-Global Block Matching
- etc...



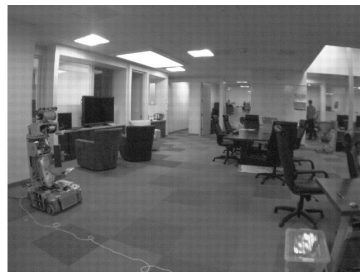
disparity



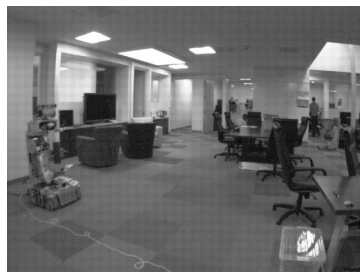
pointcloud



Let's wrap CUDA impls (with identical interfaces)



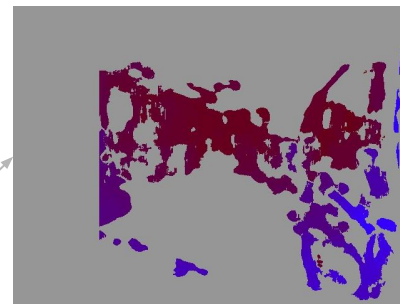
左目画像



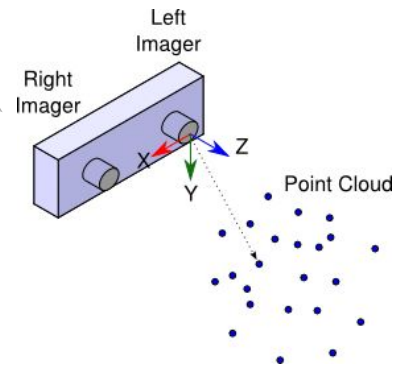
右目画像

gpu_stereo_image_proc
- Semi-Global Block Matching
+ libSGM (Fixstars)
+ VisionWorks (NVIDIA)

disparity



pointcloud



Now open to public!



https://github.com/WHILL/gpu_stereo_image_proc

A screenshot of a GitHub README file for the repository 'gpu_stereo_image_proc'. The title is 'gpu_stereo_image_proc' with a green 'PASSED' badge. The 'Overview' section states that the package provides a ROS wrapper for CUDA implementations of Semi-Global (Block) Matching (SGM or SGBM), supporting Fixstars libSGM and NVIDIA VisionWorks. The 'Requirement' section lists: CUDA (compute capabilities >= 3.5), VisionWorks 1.6 or later, OpenCV 3.2 or later, CMake 3.1 or later, and ROS Melodic. The 'Installation' section notes that only the 'Build from Source' option is provided and shows terminal commands to clone the repository and run 'catkin_make'. A note indicates that libSGM is pulled in as an external project. A final note explains how to specify CUDA architecture using build options like '-DCUDA_ARCH' or '-DCUDA_ARCH=sm_72' for Jetson Xavier.

```
cd <path-to-your-catkin-workspace>/src
git clone https://github.com/WHILL/gpu_stereo_image_proc.git
cd ..
catkin_make
# Note: libSGM is automatically pulled to gpu_stereo_image_proc/libSGM as CMake's external project.
```

```
catkin_make -DAUTO_DETECT_ARCH=OFF -DCUDA_ARCH=<your-selection>
# e.g. -DCUDA_ARCH="arch=sm_72" for Jetson Xavier
```


What is AWS IoT

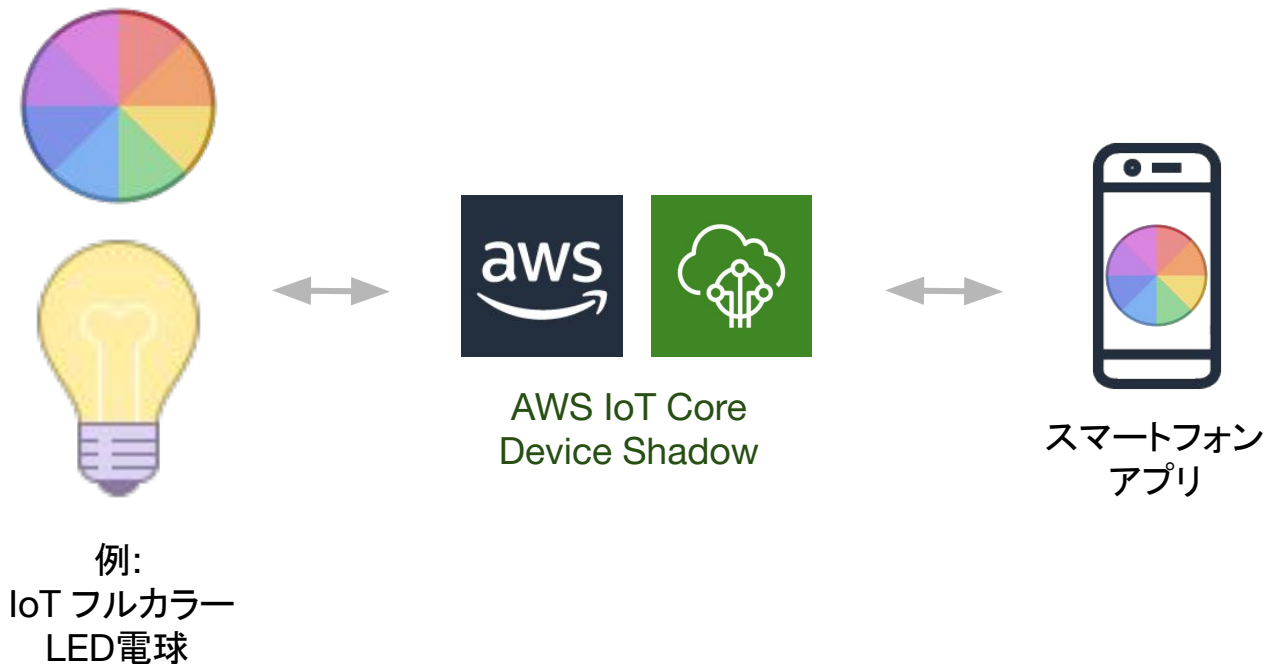


AWS IoT Core

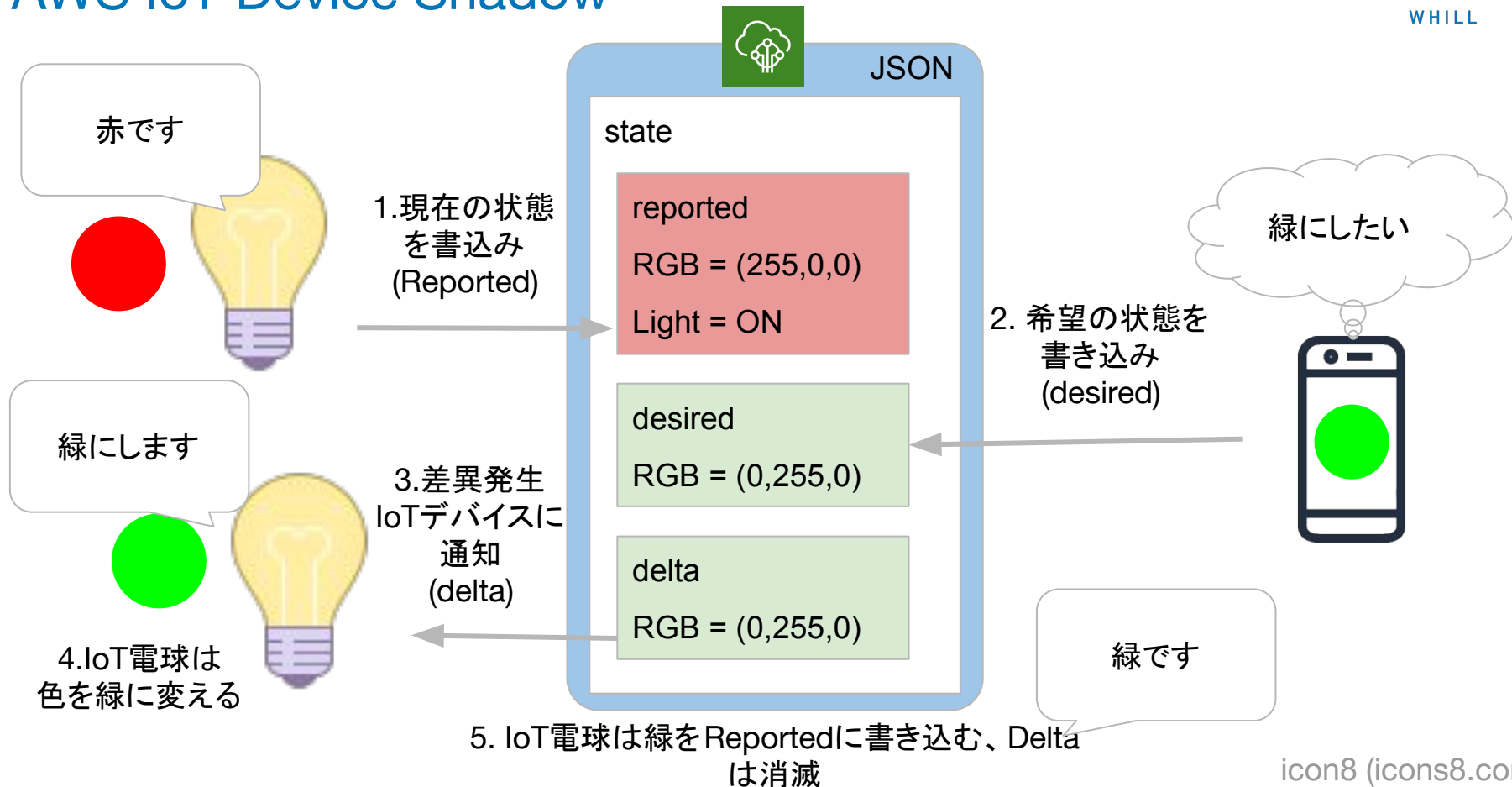
IoTデバイスとクラウドの連携をスムーズにする仕組み

- IoTデバイスからのデータをトリガーに処理
 - S3 バケットに保存
 - Lambdaで処理
- サーバレス(=自分でサーバーを持たなくてよい)で IoT デバイスの状態を保存/同期
- MQTTブローカ **Device Shadow**
- ほかに色々、AWSとの連携

AWS IoT Device Shadow



AWS IoT Device Shadow



Dynamic Reconfigure + Device Shadow



 ROS

Dynamic Reconfigure

meets



AWS IoT Core
Device Shadow

`aws_iot_bridge` パッケージ

Device Shadow + Dynamic Reconfigure



AWS IoT Device Shadow



Dynamic Reconfigure

```
"reported": {  
  "lights": {  
    "color": "GREEN"  
  },  
  "engine": "ON"  
}  
"desired": {  
  "lights": {  
    "color": "RED"  
  },  
  "engine": "ON"  
},
```

Device Shadowと
Dynamic Reconfigure
間で自動同期



- int_t
- double_t
- str_t
- bool_t

同期する内容
は.cfgに記述

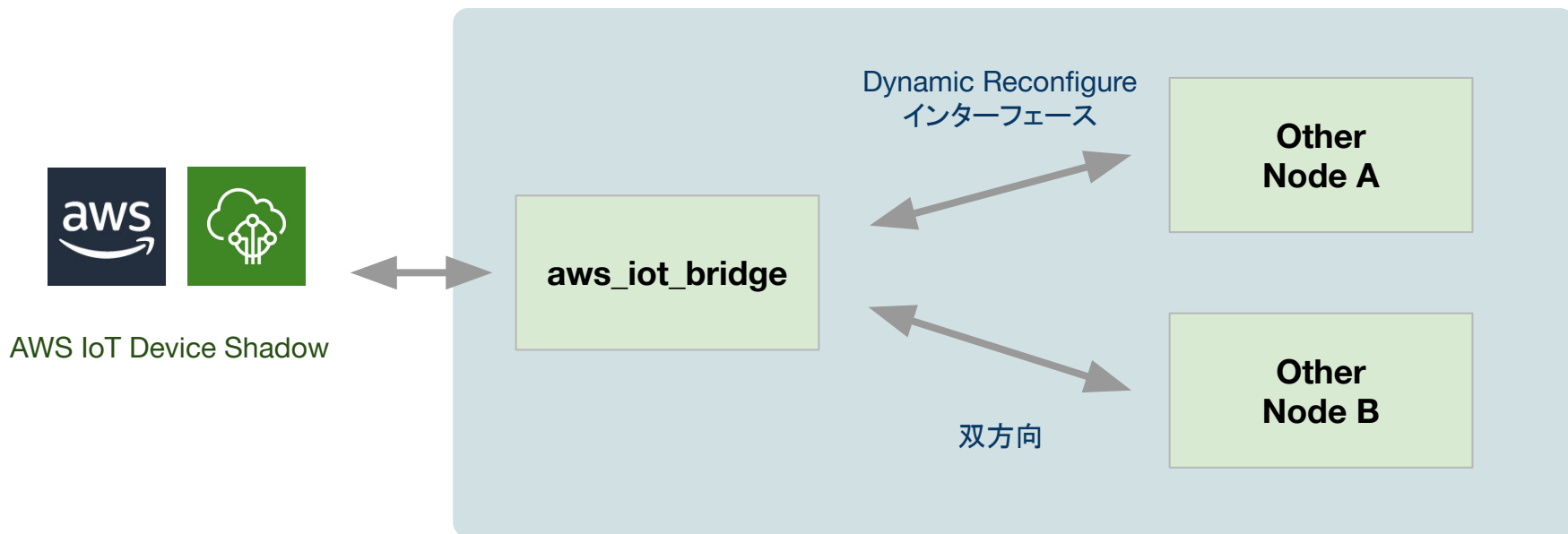
aws_iot_bridge

Device Shadow + Dynamic Reconfigure



ROS

Dynamic Reconfigure

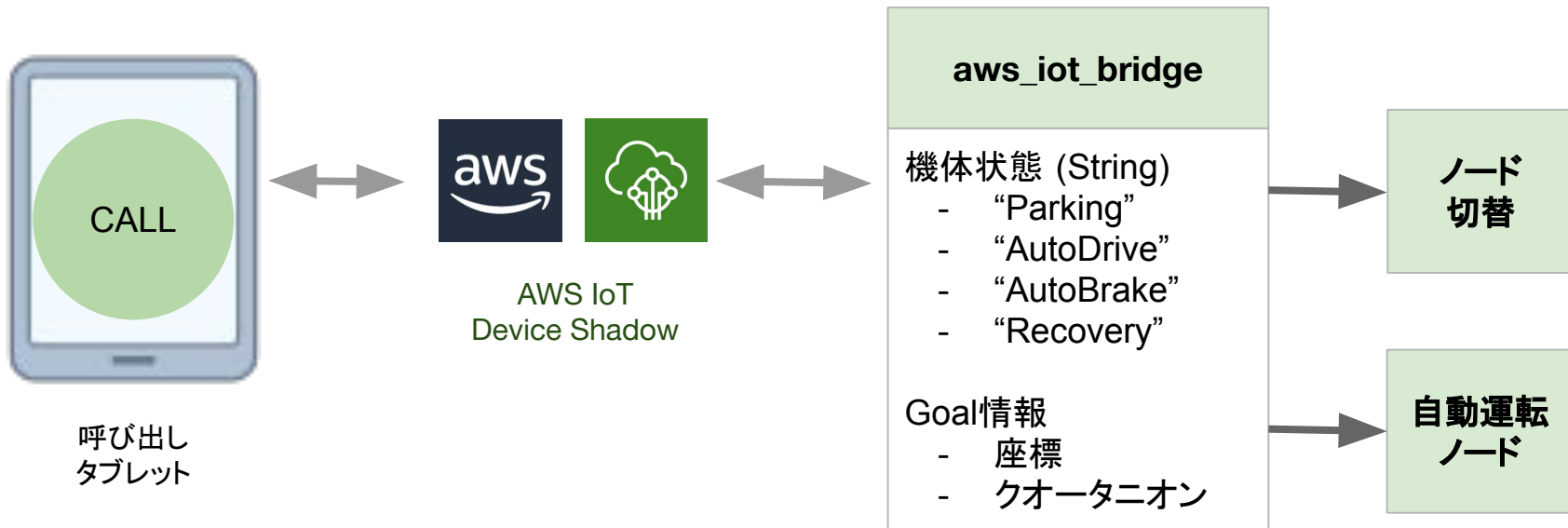


WHILL meets ROS at CES 2019!



<https://youtu.be/JVf-OTqWyKM?t=11>

aws_iot_bridge in CES



Not open to public.... :(



`aws_iot_bridge` パッケージ

近日公開予定
しばらくお待ち下さい

We're hiring!



サーバーエンジニア/ モバイルアプリ エンジニア
募集中!



WHILL

Design your own road