

# ロボットや自動車等の自動運転を実現する 「クアッドキューブ全位置検索技術」のご紹介

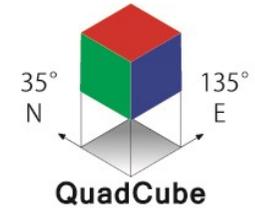
クアッドキューブ株式会社

代表取締役 畑中 豊司

2017年5月27日

# 自己紹介

つくばチャレンジの参加ロボット。左は筑波大学・右は明治大学



## 代表者略歴

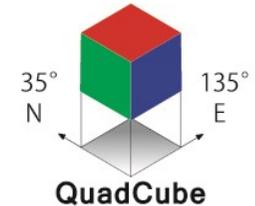
1984年3月  
1984年4月  
1999年3月  
1999年9月  
2017年 2月

代表取締役 畑中豊司(1961年4月30日生 大阪生まれ)  
在住:メイン 京都府長岡京市。サブ 東京都大田区大森町(月7)。  
滋賀大学教育学部 卒業 (教育心理)  
立石ソフトウェア株式会社 入社(現オムロンソフトウェア)。  
オムロンソフトウェア株式会社 退社(勤続15年)  
有限会社データ変換研究所 創業。代表に就任中(勤続18年超え)  
クアッドキューブ株式会社 設立。代表を兼務。

## つくばチャレンジ歴

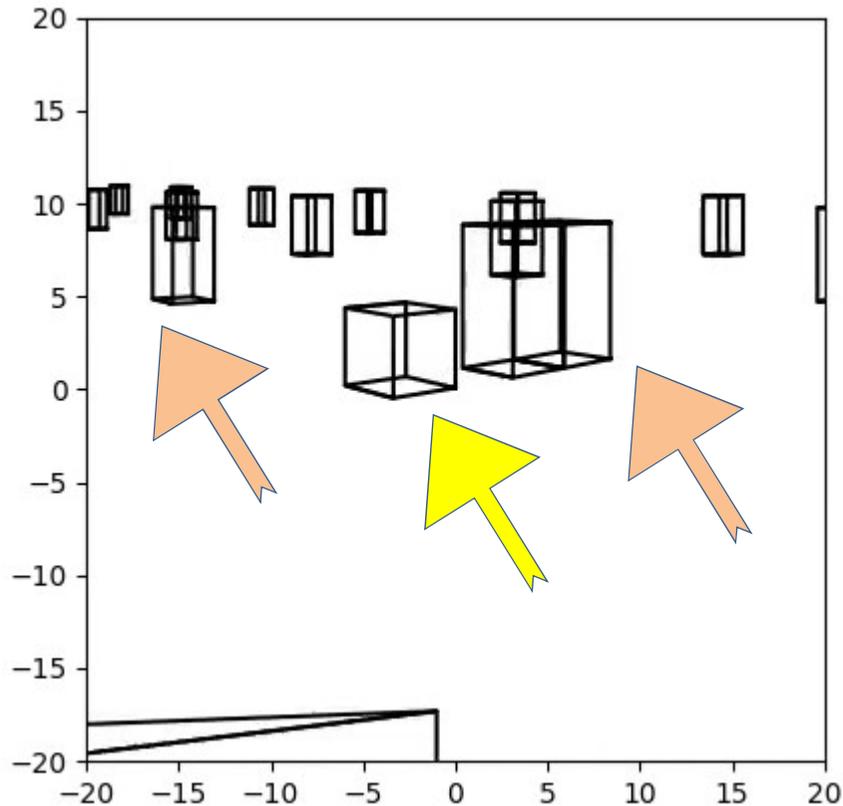
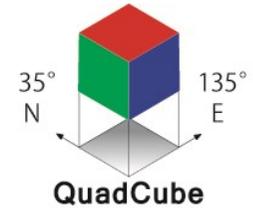
2011年 1月 ArduinoをUSBでAndroid に接続するM2M環境。Derimo 開発開始  
2012年12月 Derimo for Education を発表  
2013年11月 つくばチャレンジ2013 では OIT様 チームの一員として参加。  
2014年 7月 つくばチャレンジ実行委員を拝命。モニタリング実施を表明。  
2014年11月 Derimo TC2014 にて、23 ロボットの登録。位置と方向を表示。  
2015年11月 Derimo TC2015 にて、14 ロボットの登録。  
2016年11月 Derimo TC2016 にて、13 ロボットの登録。  
2017年11月5日 Derimo TC2017 にて、モニタリング環境を提供する予定(本走行)。  
(つくばチャレンジの実験走行 7/8, 9/23 10/14, 10/15, 10/29, 11/3, 11/4 )

# 会社紹介



商号	クアッドキューブ株式会社
資本金	100万円（株主 株式会社データ変換研究所 100%）
創業・設立	2017年2月3日(平成29年)
本社所在地	〒144-0052（最寄り駅 JR蒲田駅、京急蒲田駅 徒歩6分） 東京都大田区蒲田4-42-3-201 BIZ Kamata
TEL	03-6715-8404
FAX	03-6715-8405
E-mail	quadcube@dehenken.co.jp
URL	<a href="http://www.derimo.net/">http://www.derimo.net/</a>
経営理念	お客様にベストフィットを提案する リソースの管理と健全性の維持 リソースモデルの企画と、リスクと機会を評価する
標語	バーチャル立方体で世界をみる、観自在。
ミッション	緯度経度四辺形(LLQuad)によるクアッドキューブで計算量を軽減 させ移動体が安全で安定した走行・飛行ができるリソースを提案
ロゴ	QuadCubeロゴ3色は次を示す ベストフィット提案 / リソース管理 / リスクと機会の評価

# クアッドキューブのデモ



1つのキューブはビルディングを示す。

中心建造物 ( $x,y,z=10,10,10$ ) (2階建)。



ビル建造物 ( $x,y,z=10,10,20$ ) (4階建)。



中心建造物に視点を合わせ、  
360° を1° ずつ、  
視点を中心に回転。

パイソンの matplotlib でMP4 動画化

1° を1フレーム

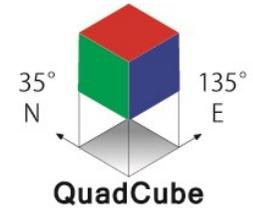
1フレーム30msec

1回転 360フレーム 時間10.8秒間。



demo.mp4

# LLQuadとは



## 構造物の抽象化表現

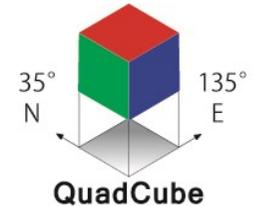
本ソフトウェアのLLQuad (Latitude longitude quadrilateral: 緯度経度四辺形) の構想が生まれたのは2015年6月ごろ金沢市の香林坊を歩いているときに思いつきました。

移動体の走行中に障害となる構造物を全位置検索にて検索しようとするとき、**建物の長さや方向が一様でないので表現しにくい**という問題がありました。

この解決方法として、**緯度経度の線に平行な占有格子を10km、1km、100m、10m、1m、10cmで表現**します。

LLQuadのタイプ	定義値	対応する度数	例	
LLQuad10km	1	0.1	35.1	LLQuad10m (定義値4)は、およそ10m×10mの大きさです。正確には緯度により異なり、京都駅付近では東経135.7594度、北緯34.9876度のとき11.1m×9.0m。緯度が高い札幌駅(141.3507,43.0687)では経度辺が短くなり、11.1m×8.1mとなります。
LLQuad1km	2	0.01	35.12	
LLQuad100m	3	0.001	35.123	
LLQuad10m	4	0.0001	35.1234	
LLQuad1m	5	0.00001	35.12345	
LLQuad10cm	6	0.000001	35.123456	

# クアッドキューブとは



クアッドキューブ (QuadCube) は、構造物データの簡易な立方体で表現する方法 LLQuad (緯度経度四辺形) に高さの情報を加え、立方体の表現にしたもの。高さは地表高、立方体底面高、立方体高の3つ。京都タワーのイメージで131mの高さの構造物は、CSVで  
Q4,135.7594, 34.9876,0,0,131

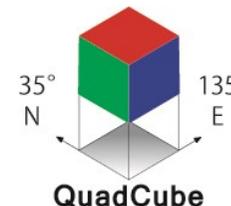


LLQuad ( Latitude Longitude Quadrateral ) 面のCSV

Q4 行頭文字Qはシンプルなクアッドキューブ。行頭識別文字2バイト目は1から7あり、LLQuadのタイプを示す。続く位置情報が全位置検索の対象。  
xlon, ylat 位置情報 (xlon, ylat = Q4,135.7594, 34.9876) は全位置検索の対象位置。  
高さ 高さは、地表高・立方体底面高・立方体高(0,0,131)を順に指定します。

クアッドキューブの位置情報はインデキシングされ、全位置検索の対象となります。行頭識別文字の先頭は、アルファベットが大文字(この例ではQ4)のとき、本ライブラリが使用し、アルファベットが小文字のとき(例えば「q4」)、ユーザ定義情報となります。CSVは、1バイト目が大文字がシステム・小文字がユーザ、2バイト目が数字の時、LLQuadデータとして扱い、2カラム3カラム目をxlon, ylatとし、4-6カラム目を高さとして処理します。全位置検索時に位置情報を与えると、検索結果にとして得ることができます。

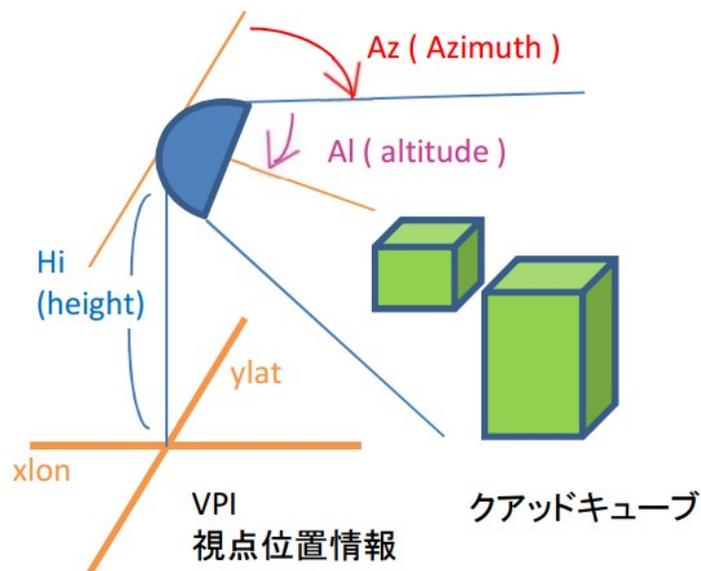
# VPIとは



視点位置情報 (VPI: Viewpoint position information)

VPIは、モニタリングで眺めるときの目線の位置情報を定義するものです。これにより、どの緯度経度から、どの高さで、どの方向に、どの目線の向き(仰俯角)で、風景を眺めるのかという目線(例:カメラの角度)を定めることができます。

VPIは、モニタリングの視点を定めるデータ構造体をいい、位置情報(xlon, ylat)、hi (height: 高さ)、az (azimuth: 方位角)とal (altitude: 仰角・俯角)で構成されます。



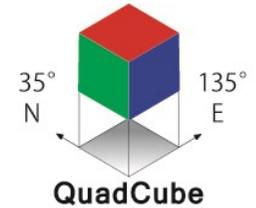
VPI視点位置情報より参照

方位角: (英語: azimuth (アジマス)、略記号 az)は、本ソフトウェアでは、右手系東基準として方位を扱い、東を $0^\circ$  北を $90^\circ$  とします。

左手系北基準の北を $0^\circ$  東を $90^\circ$  とする場合、フラグで指定します。

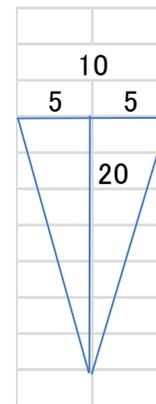
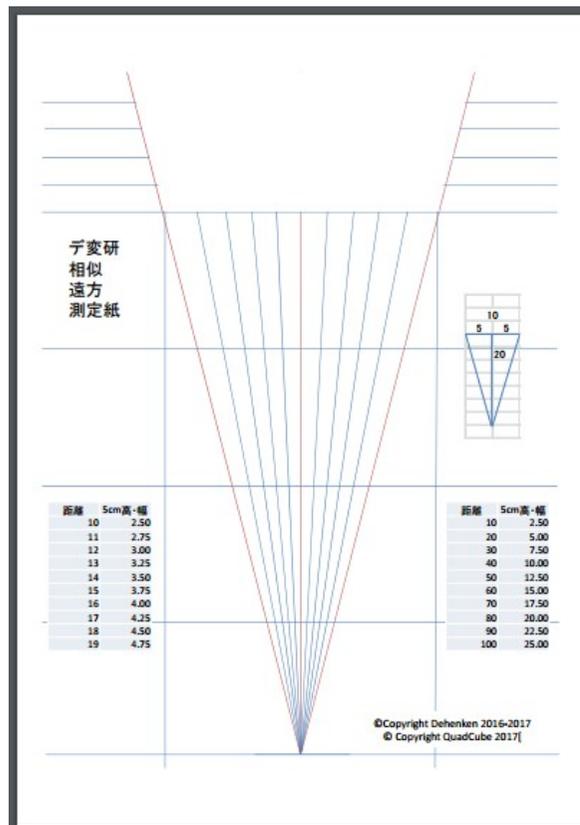
仰俯角: 仰俯角(ぎょうふかく、英語: altitude (アルチチュード)、略記号 al)は、水平を基準とし上下方向の角度を指定します。上向きは仰角(ぎょうかく)で、真上は $90^\circ$ 、下向きは俯角(ふかく)で真下は $-90^\circ$  です。

# 紙のVPIの説明



## 視点と視角と距離と長さの関係

お手元に視点と視角と距離と長さの関係を目視確認できる用紙を用意しました。  
 構造物の長さや幅が、距離(距離は長さ幅)に相似するというを確認できます。

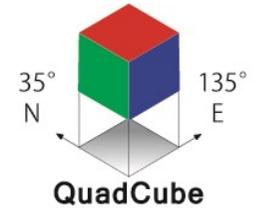


距離	5cm高・幅
10	2.50
20	5.00
30	7.50
40	10.00
50	12.50
60	15.00
70	17.50
80	20.00
90	22.50
100	25.00

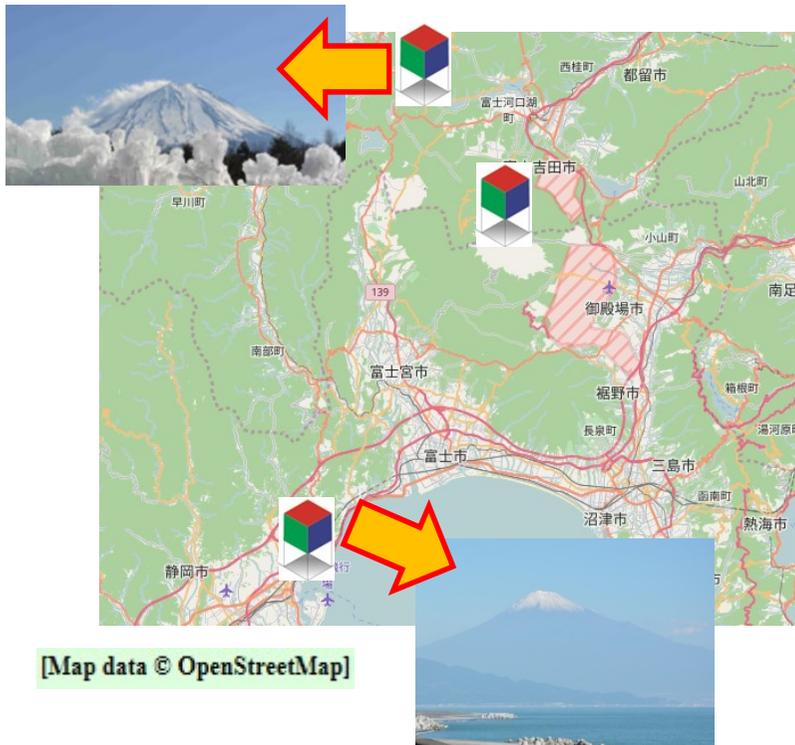
距離	5cm高・幅
10	2.50
11	2.75
12	3.00
13	3.25
14	3.50
15	3.75
16	4.00
17	4.25
18	4.50
19	4.75

- ※ 本用紙は、距離と高さや幅の関係を  
確認をするためのものです。  
ご使用時は目に近づけすぎないように、  
(直接、眼に触れないよう)注意して下さい。
- ※ 本用紙で5cmの部分の視角は14.04度です。  
視角を20度にするには7.28cmにします。
- ※ 人の視野は水平60-90度、垂直45-70度程度

# 距離と大きさ



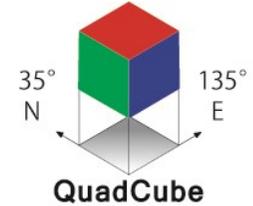
見えている構造物の大きさは相似で  
富士山(標高3.776km)を、  
西湖野鳥の森公園(18km)から見るのや、  
三保の松原(43km)から見るのを、



高さ4cmの「ラーメン横綱」の餃子一人前の  
無料券(10枚綴り)で作った山で似せると、  
30cmのものさしの18cmや43cmの位置  
を参考にすると、類似画像が得られます。



# LVAとは



## 格子窓座標 (LVA: Lattice View Angle Coordinates)

格子窓座標は、格子窓(障子(しょうじ)の棧(さん))を座標位置に見立て、視角 $\pm 20$ 度の上下左右の範囲を、描画位置 $\pm 20$ の座標に描画させる座標系。

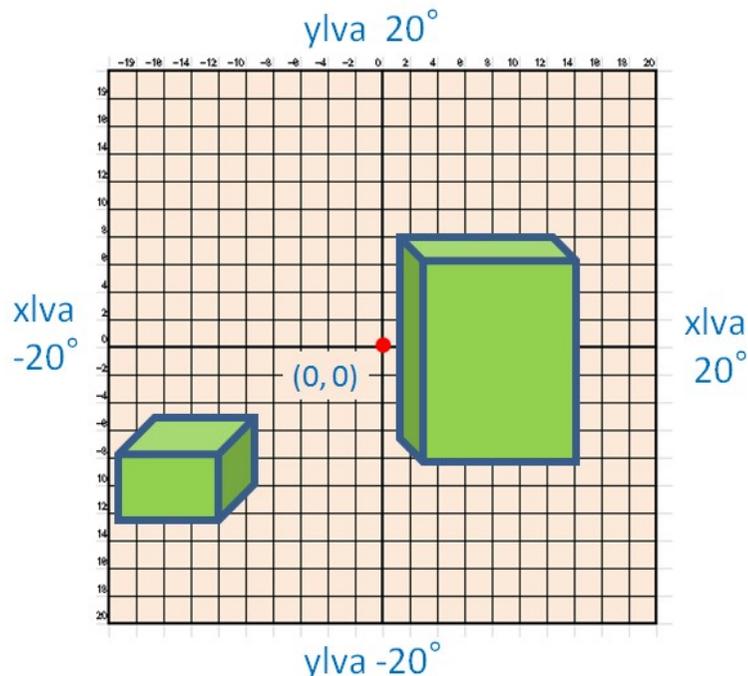
座標の(xlva, ylva)の1.0 LVAは視角1.0度と一致。

LVA の中点(0, 0)は、VPIで指定した視点方向と一致。



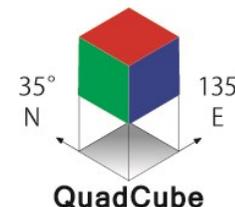
demo.mp4

デ変研QCライブラリは、現在位置に近いクアッドキューブを絞り込み、描画情報(このLVA)を返すことで、容易に視野画像を再現させるプログラムです。



- ・ あらかじめ構造物を大量のクアッドキューブのデータで作成した電子地図情報を用意します。
- ・ ユーザが必要な視野位置情報(VPI)を与えます
- ・ ライブラリはクアッドキューブを全位置検索します。
- ・ ライブラリはCube構造体に立方体の8点のLVA位置を計算済みにして返します。
- ・ ユーザは立方体8点の位置を、12本の立方体の稜線として2Dの描画ウィンドウに線で描画します。
- ・ これだけで容易に2D風景画を再現できます。

# ちなみに視力とは



視力とは、目で物体を識別できる能力のことです。

## ランドルト環

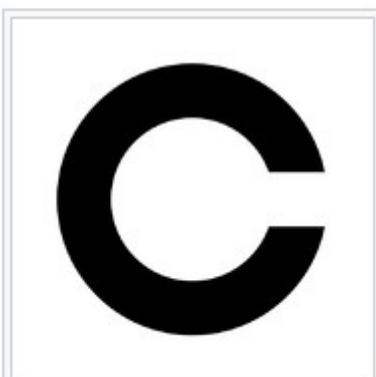
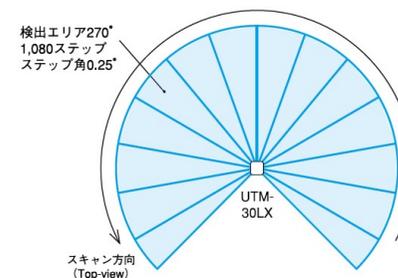
視力を測定する際に用いられる視標が「ランドルト環」です。

「一定の距離から認められる二点間の最小間隔」いわゆる「最小分離閾」を調べます。

ランドルト環の全長が75mmのとき、環の切れ目の長さは、全長の1/5で、**15mm**。

**5m**離れたところから、この切れ目が判別できたときの**視力が0.1**となります。

このことを視角で度数を表現すると、0.364度の度数を判別できたとき視力0.1です。



ランドルト環  
円環全体の直径:円弧の幅:輪の開いている幅=5:1:1の比率である。

ロボットに搭載している、「眼」となるセンサは、測域センサ(そくいきセンサ、英語: Laser Range Scanner / Lazar Range Finder / 3D scanner / URG (Hokuyo) )といいます。

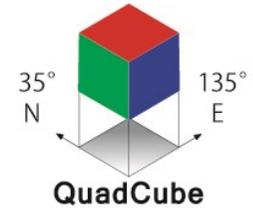
角度分解能としては、様々ですが、たとえば、ステップ角:約0.351° (360° /1024分割)の場合、0.1程度で、この機械は0.1の視力と関係が似ています。右のTOP-URGでは、0.25° です。

※ 分解能を高くすれば視力が向上します。しかし、空間識別のための計算量が増え、処理する時間も増えます。



北陽電機株式会社  
UTM-30LX  
【通称:TOP-URG】  
検出エリア270°  
1080ステップ  
ステップ角0.25° 走査時間25ms(モータ回転数2400rpm)

# SODA(操舵)データ



## Sodaデータを移動体の速度との関係

Soda(操舵)データについて構想が生まれたのは2016年1月ごろのことです。LLQuadで10cm四方のデータで日本全体を埋め尽くすとデータ量が多くなり過ぎます。データを間引くことは必要でしたが、データ量をどの程度抑えるのか不明でした。

自動車のハンドルを切る方向を10秒先まで知りたいというニーズがあり、そこからSoda点は移動体の1秒間に進む距離とするという考えができました。

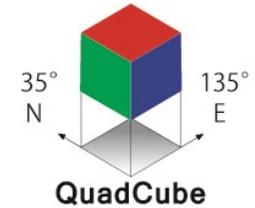
10秒先のハンドルの操舵方向は10点を先読みすればわかります。

高速道路を時速100kmで走行するとき移動体の1秒の移動距離は27.778mとなり、東名高速道路の全長は346kmで片道のSoda点は12,456個となり、これぐらいであれば情報量は軽微といえます。

右図は、イメージ画像で、高速道路に一定の間隔で切れ目を入れています。



# 全位置検索



大量に作成されたクアッドキューブの地図データを、あとから検索しやすいように、インデックスを作成して、群にまとめて格納します(インデキシング)。

移動体の現在いる、位置情報と視点の方向(VPI)を与えます(インプット)、

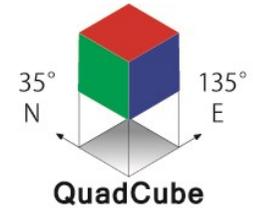
移動体の目線から可視範囲のクアッドキューブの地図データを得ます  
可視状態を再現するLVAの位置情報の計算結果を得ます。(アウトプット)

LVA情報と移動体の目のセンシング結果と突合せます。

- ① 移動体が自己位置と認識している情報との比較(自己位置推定)
- ② 移動体の進行方向(Soda)の確認(自律移動)
- ③ LVAデータを2D上に描画し、それを人間が目視して確認・監視(妥当性確認)

以上のように、移動体の自律移動に活用できるものと考えています。

# 全位置検索のAPIと手順



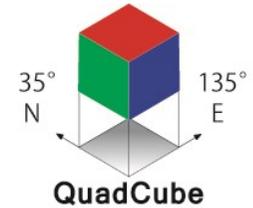
デ変研QCライブラリには、APIが用意されています。大量・大容量のクアドキューブの構造物情報を装着していても、必要な位置情報の周辺データだけを高速に検索実行することができます。

初期処理	<code>int Dhkqc_init(char *fn, int flag, int stage, int params);</code>
装着処理	<code>int Dhkqc_setqc(unsigned char *buf, int size, int flag);</code>
加工処理	<code>int Dhkqc_hash(int flag);</code>

検索処理	<code>int Dhkqc_cube(int flag, vpi_t *vpi, cube_t *cubep, int ncube);</code>
状態変更処理	<code>int Dhkqc_one_second_later(int flag);</code>
直接読出し処理	<code>int Dhkqc_get_data(int flag, 構造物, int no);</code>
直接書込み処理	<code>int Dhkqc_put_data(int flag, 構造物, int no);</code>

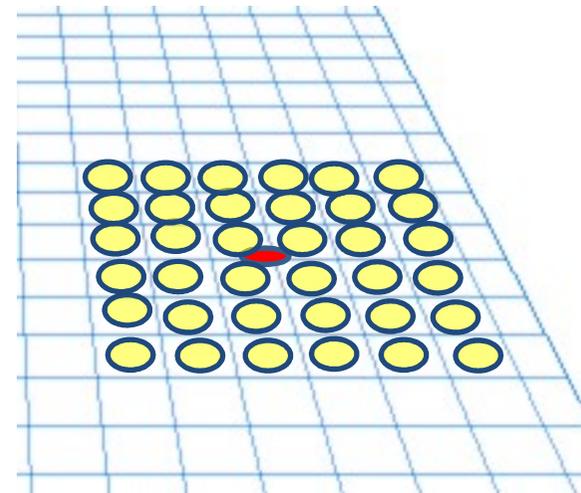
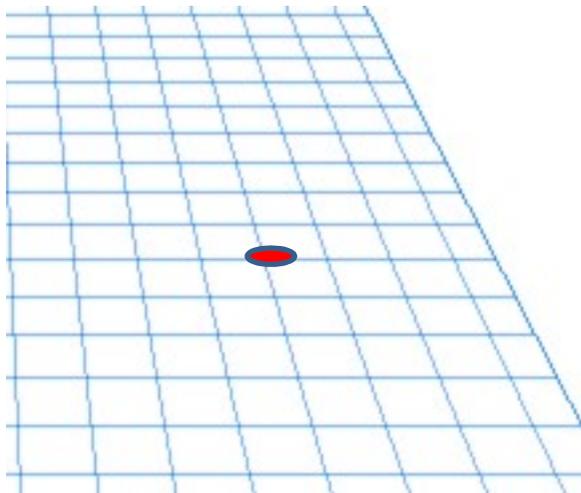
終了処理	<code>int Dhkqc_term();</code>
------	--------------------------------

# パネルサーチ (Dhkqc\_cube関数)

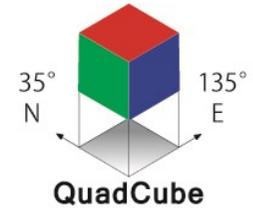


パネルサーチとは、全位置検索にて、インデックスの検索を実施するとき、指定された位置のエリアを指定された目の数のプラスマイナスの平方数で、インデックス内に格納されているデータをもれなく取り出す方式です。

下の左図は、指定する位置を赤丸で表し、  
下の右図は、パネル数 3 を指定した時に -3から+3まで、  
最終的に 6×6 のパネルを検索をしていることを示します。



# 移動体オブジェクト



移動体オブジェクトとは、  
本ソフトウェア内で、指定する移動体の四角い形状の物体をいいます  
移動体オブジェクトは、  
与えられた操舵(Soda)データに従って、  
与えられた移動体コマンドにより指定された速度で  
移動シミュレーションをします。

```
1 |#ISD 1.0↓
2 |# Dehenken Derimo Series : LLSoda data↓
3 |# (C) Copyright Dehenken Limited 2016-2017.↓
4 |# All rights reserved.↓
5 |#↓
6 |#↓
7 |# Mobile Object: MO,x,y,z,az,second,times,nam
8 |MO,3,20,4,90,10,5,Mover No.1↓
9 |#↓
10|# soda data↓
11|#↓
12|S6,135.757000,35.000000,0,S,20↓
13|S6,135.757000,35.003000,0,S,20↓
14|S6,135.757000,35.004000,0,S,20↓
15|S6,135.757000,35.005000,0,S,20↓
16|S6,135.758000,35.006000,0,S,20↓
17|S6,135.759000,35.006000,0,S,20↓
18|S6,135.759000,35.006000,0,S,20↓
19|S6,135.759000,35.005000,0,S,20↓
20|S6,135.759000,35.004000,0,S,20↓
21|S6,135.758000,35.003000,0,S,20↓
22|S6,135.757000,35.003100,0,S,20↓
23|S6,135.757000,35.003000,0,S,20↓
24|[EOF]
```

LLSoda ( Latitude Longitude Soda data ) 面のCSV

MO: 移動体オブジェクト 例) MO, 2,4,6, 90,10,5

MO (Mobile Object) は、xyzの大きさ(単位m)で移動体オブジェクトの形状を指定します。

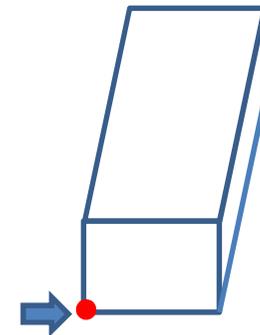
S6: Soda点 例) S6,135.759227,35.004831,0,Command

行頭識別文字はS6 で、1つのSoda点を指定します。

位置情報(xlon, ylat)、地表高、移動体コマンドからなります。

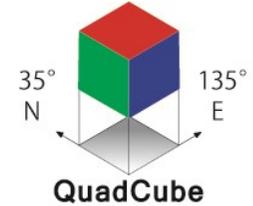
CM: 移動体コマンド行 例) CM,S,10

移動体オブジェクトは、本ソフトウェアの構造物情報の中を動きます。  
本ソフトウェアを通じて様々なモニタリング位置から見ることができ、  
現実世界を仮想的な移動風景を作り、シミュレーションできます。



上の図は、移動体オブジェクト(x,y,z = 5,20,5)の例です。  
これは、電車の車両型オブジェクトでこの車両がSoda情報に従って走行シミュレーションします。  
(現在、開発中)  
赤は基点で(xyz)=(000)です

# 移動体コマンド

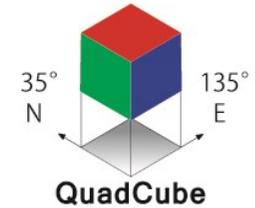


移動体コマンドとは、移動体オブジェクトがSoda位置からそこにあらかじめ与えられた情報を読み込んで、運航のための指示を移動速度に変容させるものをいいます。

運行管理者は、移動体オブジェクトがその位置にいた場合の最適な速度を設計し、移動体コマンドにて、移動体オブジェクトに変容を与えるために設定をします。

コマンド	コマンド内容の詳細
S / Speed, n	時速 n kmと指定し、加速(あるいは減速)します。Soda方向は次のSoda点を目指します。
W / Wait, n	停止時間(待ち時間)は無操作となる時間を指定します。
D / Destroy, n	オブジェクトが消滅します。nは消滅までの秒数です。
C / Create, m, xlon, ylat	クリエイトをします。m は生成までの秒数で、xlon, ylatの位置に生成します。
DC / DestroyCreate n, m, xlon, ylat	オブジェクトを n 秒後に消滅させ、m 秒後に xlon, ylat の位置に生成させます。
G / Goto, label	labelは任意の文字列の指定が可能で、LBの行にて指定されたSoda点がリンクし、Soda線を形成します。これにより繰り返しループさせることができます

# 移動体オブジェクトの慣性情報



移動体オブジェクトは慣性情報を保持します。  
慣性情報とは、現在移動速度と現在移動方位です。

移動体オブジェクトは、次に与えられたSoda点に向かって移動しようとし、移動体コマンドの速度指示や全位置検索の結果、標識情報の指示を知得。移動をしていくとき、慣性情報として現在移動速度と現在移動方位を内部で設定。

Soda点の喪失は、急な曲がり方を指定されたとき、鋭角を曲がり切れなかったとき、または、停止命令がないまま、最終のSoda点を行き過ぎたとき。

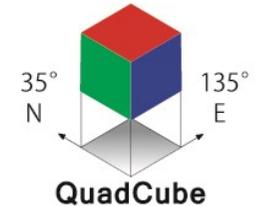
移動体オブジェクトは慣性情報に従って、等速直線運動で移動行動を継続。移動体オブジェクトがSoda点を喪失したとき、移動体は自己の制御の設定で、

- ・停止をするか、
- ・慣性情報による移動を継続し、なんらかの停止行動に至るか、

あらかじめ、移動体のSoda点喪失ポリシーを設定する必要があります。

ドローンの場合、Soda点喪失時に、帰着ルートと帰着地点を用意するなどします。

# 標識オブジェクト



標識オブジェクトは、移動体オブジェクトに行動に変容を与える指示体で、位置情報を固定し、その場所に設置されるものです。

この標識オブジェクトは、2017年4月ごろ、本「QCコマンド開発仕様書」を作成時に LLQuadとLLSoda面とで移動体が同時に移動できる仕様を考えたとき、移動体同士が接触する場合があります、これを回避するために、標識オブジェクトの考え方が誕生しました。

標識オブジェクトは、4つです。

交差標識(信号機)・一時停止標識・方位標識・方位速度標識

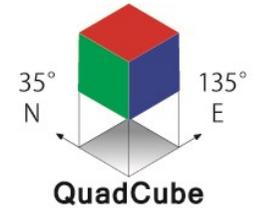
移動体オブジェクトはこれら標識を自律認識します。

移動体オブジェクトは、自律減速もしくは自律停止もしくは方位変更します。

移動体オブジェクトは、あらかじめ指定された方式で、回避行動をとるときもあります。

移動体オブジェクトは、自主判断し自律で行動をします(行動するようにさせます)。

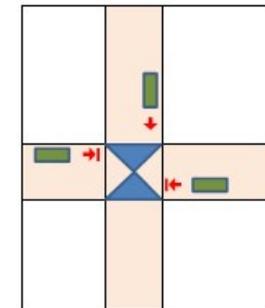
# 交差標識(信号機オブジェクト)



標識オブジェクトの1つに信号機オブジェクトがあります。

信号機オブジェクトには、通過を示す、通過標識の表現形状(スタイル)があります。2つのスタイルがあり、スクエアスタイル(S1、S2)とクロススタイル(X1、X2)です。

2つは、以下のような、四辺形に与えられた図柄で、表現し、青色のとき信号青。



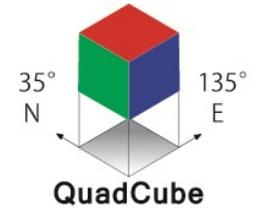
東西南北の交差点では、クロススタイルが有効で、  
45° 傾いてに交差するとき、スクエアスタイルが有効です。

移動体オブジェクトは、走行方向の±20度の範囲に交差標識を検知したとき、その通過時に、標識の青色の部分を検知した場合は「青信号」で通過できます(走行を継続します)。

移動体オブジェクトが、青色の部分を検知していなければ「赤信号」で停止します。

移動体オブジェクトは交差標識の再検知を繰り返し青色を検知すれば通過できます。

# MOFEモニタ



MOFEモニタは、マップ(M)・オーバービュー(O)・フォロー(F)・モバイルアイ(E)の4つの状態(State)をカメラの目線で移動しながらモニタリングをすることができるプログラムです。

MOFEモニタ部はプログラムを開示しています。

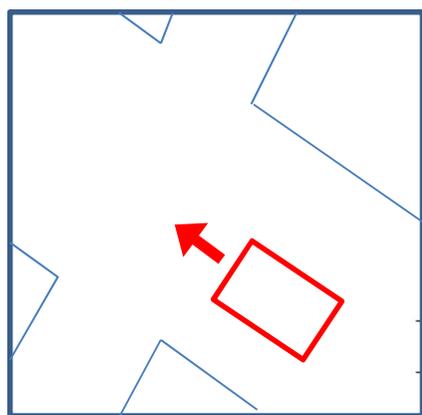
デ変研QCライブラリはライブラリ形式で提供します。

QCコマンドとデ変研QCライブラリは、Terminal上で動作します。

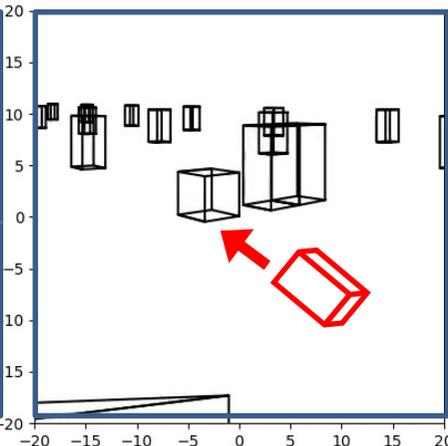
これより線画出力し、動画データ(MP4)にすることができます(後述、matplotlib)。

MOFEモニタは、つくばチャレンジ時期には、JavaScriptで実装完了している予定です。

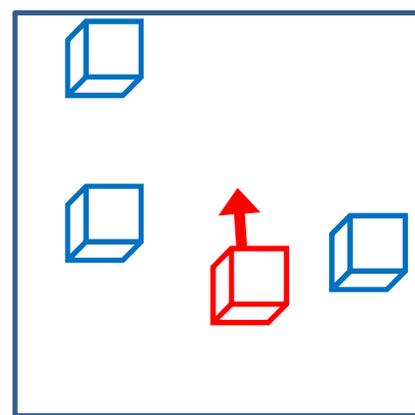
マップ(M)



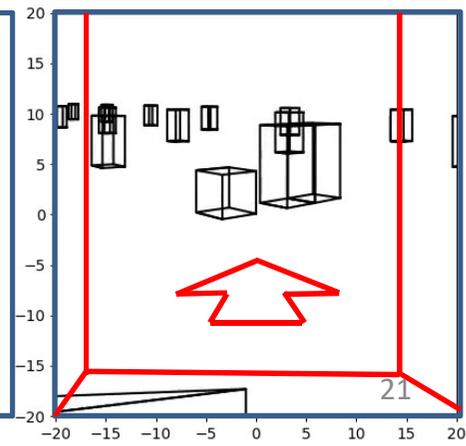
オーバービュー(O)



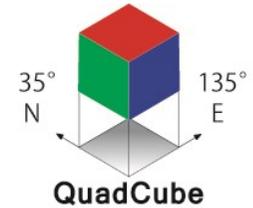
フォロー(F)



モバイルアイ(E)



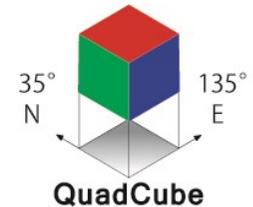
# 連携使用しているソフトウェア



本ソフトウェアと連携して使用している、ソフトウェアをご紹介します。

- matplotlib                      動画作成に使用(OSS)
- Gazebo                              ロボットシミュレータ(OSS)
- Derimo                              M2M通信環境の実現(当社開発)。
- Open Street Map                  無償地図サーバでアイコンで現在位置表示(OSS)。
- Google Earth                      Google 社によるソフトウェア(商用)。  
本ソフトウェアからKML形式に出力し、外観を確認。

# matplotlib



matplotlib は Python でグラフを描画するときなどに使われる標準的なライブラリです。画像ファイルを作るばかりでなく、簡単なアニメーションやインタラクティブなグラフを作ることも可能です。

様々な例としては、matplotlibサイトのギャラリーで確認できます。

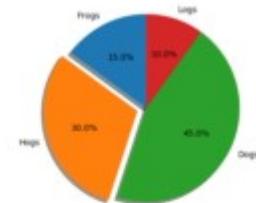
[matplotlib/gallery](https://matplotlib.org/gallery/)

当社開発の、Sample.py というパイソンで書かれたソースコードは、[www.derimo.net](http://www.derimo.net) のサイト内で OSS で公開します(2017年11月頃)。

デ変研QCライブラリで出力した線画CSV形式描画データは、1フレーム30msecのアニメーションで描画できます。先ほどの demo.mp4 が作成できます。

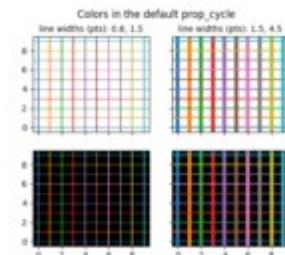
いろいろな動画による、描画サンプルも自分で作成して、最終的にこのような動画で表示することもできます。

## Pie and polar charts



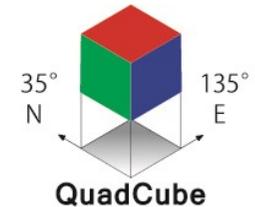
pie\_demo\_features

## Color



color\_cycle\_default

# Gazebo

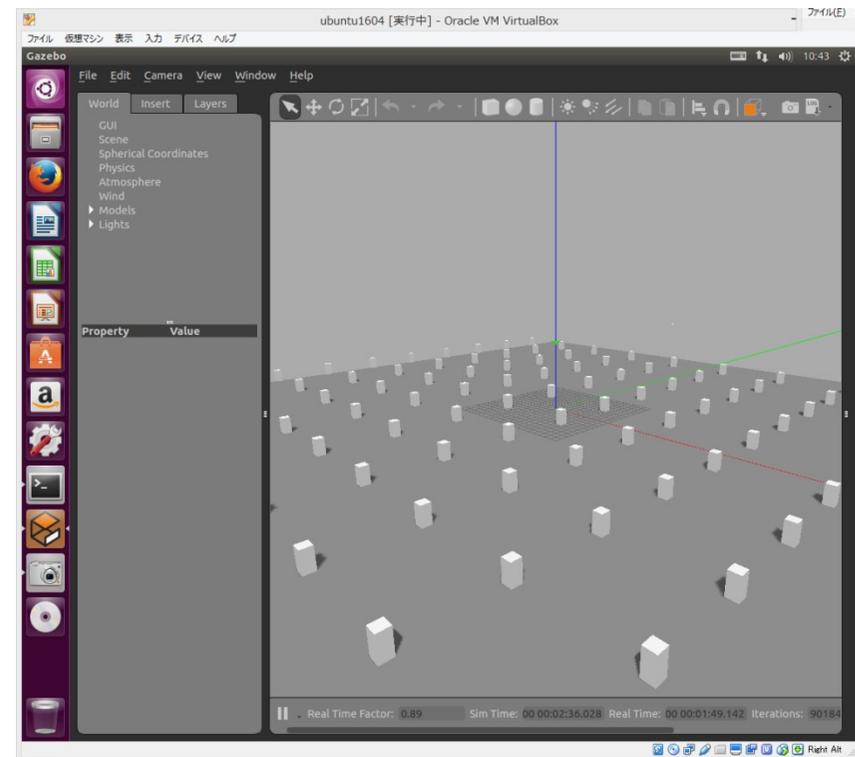


Gazeboは3Dロボットシミュレータです。特徴は強力なGUIをもち、カメラ、LRFなどのセンサシミュレーションが豊富で、ROSとの連携も充実しています。ロボットのための3Dモデルをバーチャルにシミュレートできます。

Gazeboシミュレータは 動作力学をサポートしています。現実の世界の剛体物理もロボット環境でモデル化することもできます。

右図は、クアッドキューブのデータを、**Gazebo内に、1m × 1m × 2m**のモデルで構築してみたものです。XML形式を仮作成。

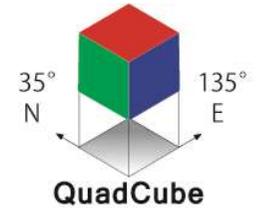
今後、クアッドキューブからのGazebo向け、XMLファイル出力機能を搭載します。



## バーチャルシミュレータ

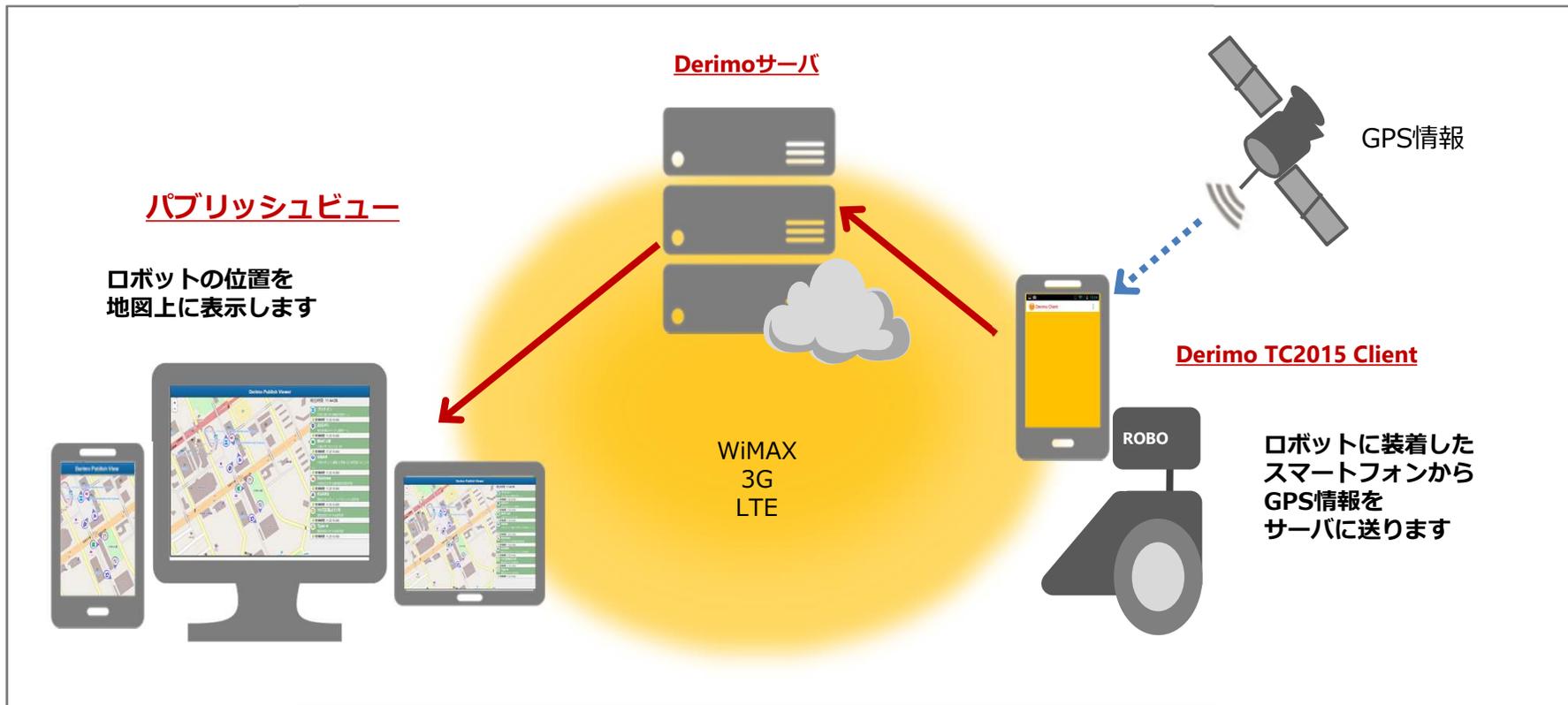
ロボットを作るためには、組み込みのファームウェアやハードウェア設計、センサーの選択、コントロール・システムの設計、そして機械設計など、様々な分野でのスキルが必要です。しかしシミュレーション環境を利用すれば、開発に高いコスト(そして時間)をかけなくても、テストや測定、ロボット・アルゴリズムの視覚化などのための仮想的な場を作ることができます。

# Derimo

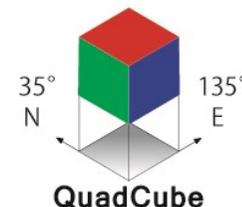


Derimoは、株式会社データ変換研究所が2015年12月にAndroidとArduinoを接続し、M2M機器間の通信を実現したソフトウェアの名称です。つくばチャレンジのモニタリングに活用しています。

## Derimo



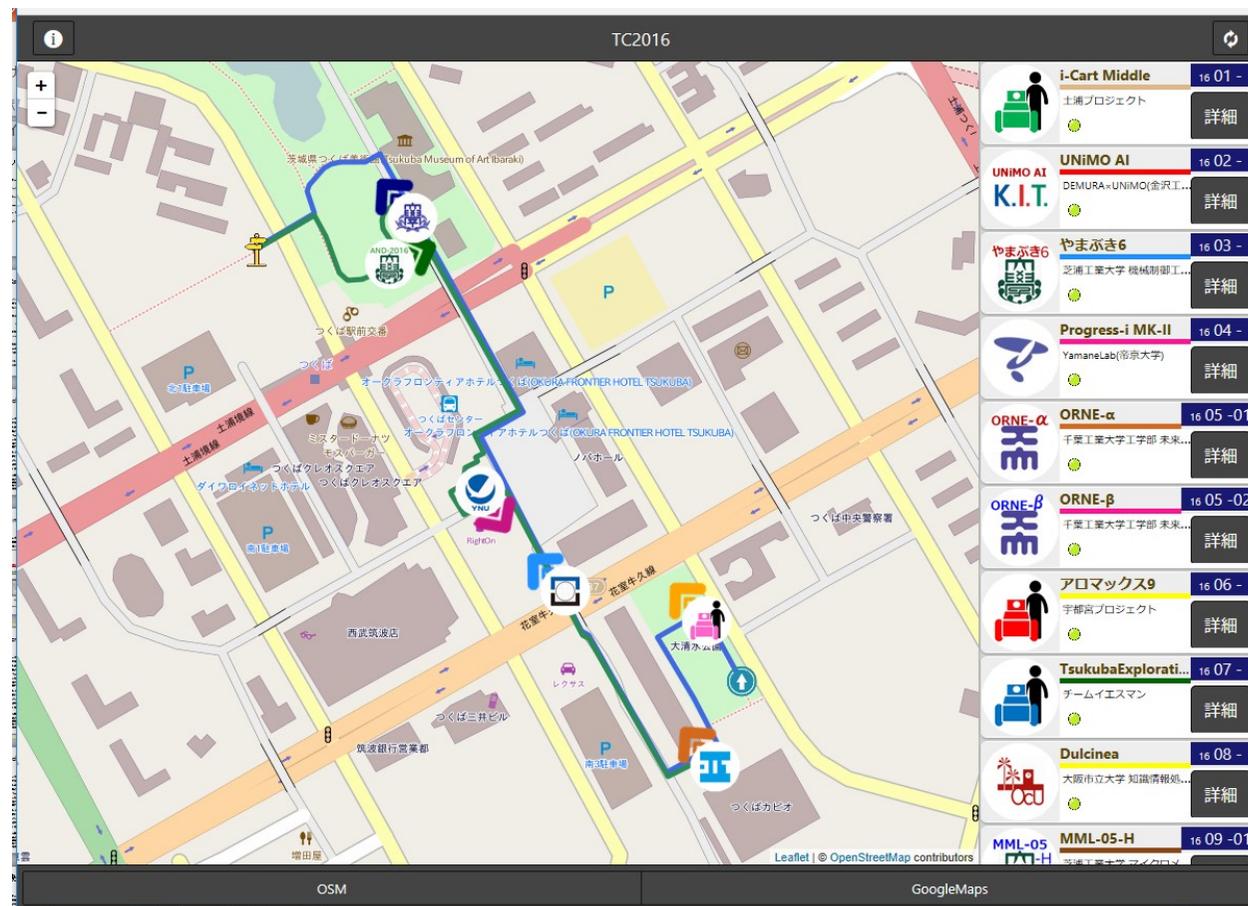
# Open Street Map



Open Street Mapは、つくばチャレンジでのロボットの現在位置のモニタリングのために活用しています。

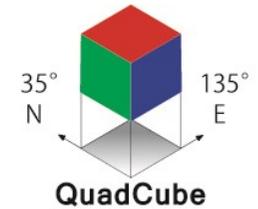
OSMサーバを弊社内で立ち上げ、閲覧している状況と、サーバの負荷をZabbixにて監視しています。

例年、その状況を監視してますが、サーバには大きな負荷はかかってない状況です。



デモ <http://tc2016.event.derimo.net/demo/>

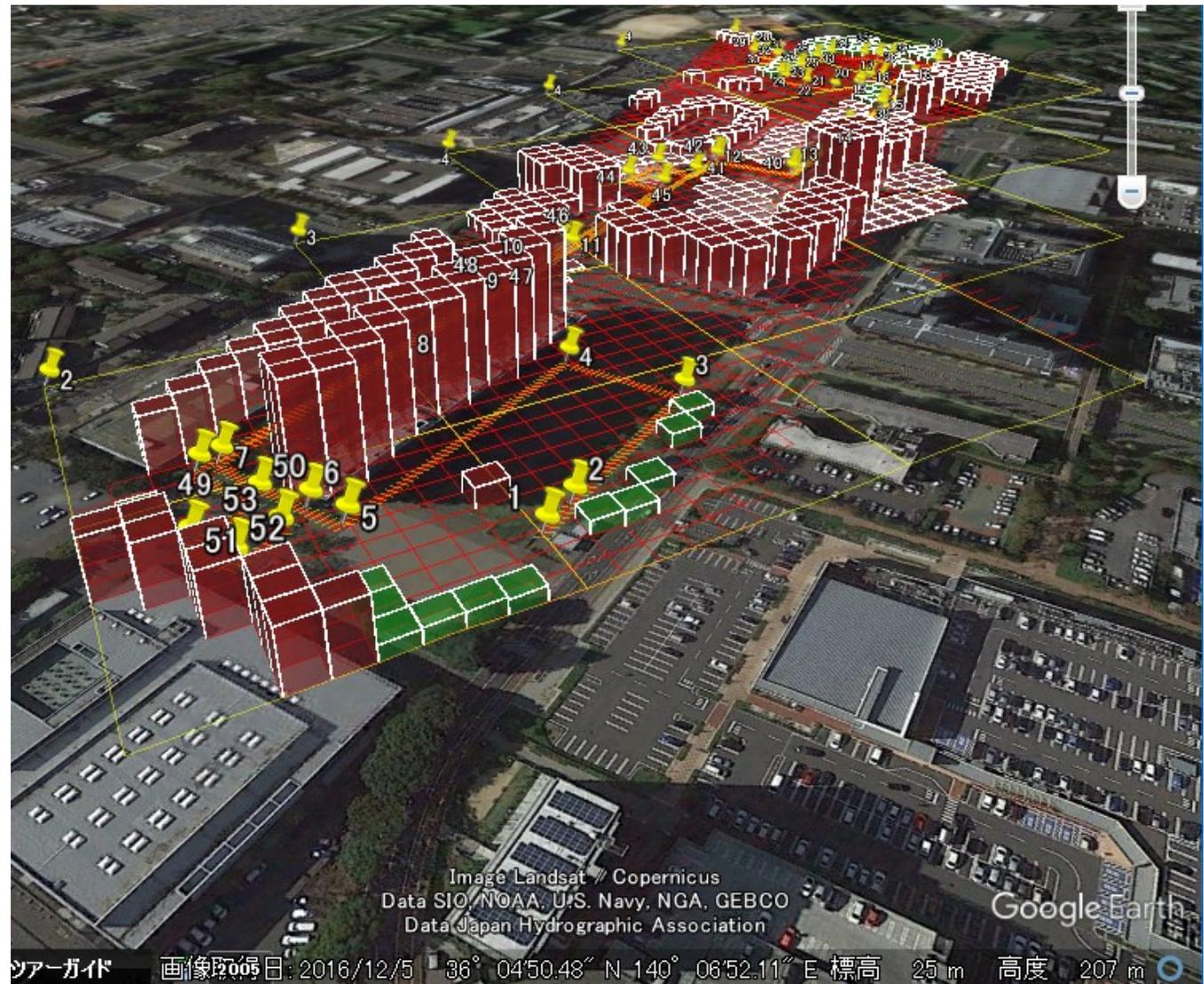
# Google Earth



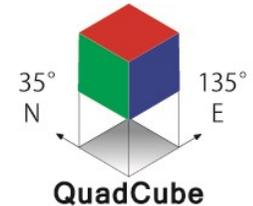
クアドキューブのデータをKMLにファイル出力することで、Google Earthで航空写真上に立方体の配置状況を確認できます。

右の表示は、つくばチャレンジのコースをKMLにしました。

- ・地表での四角域。
- ・ビルの状況
- ・SODA線の状況。



# 今後の展開



本ソフトウェアが、クアッドキューブという現実界を仮想的に観る世界を作り上げ、  
移動体の安全な自律走行を実現し、  
移動体が見えている風景と同じものを人が観て安全であると確信できる、  
このような概念の証明 (Proof of Concept) を実現しできるよう  
成長させていきたいと願っています。

移動体メーカ(自動車屋さん、あるいは自動車部品メーカ)への販売、  
地図屋さん、カーナビ屋さん、信号機屋さん、学校関係(学習用)、個人

**つくばチャレンジ2017の時期**にはもう少し完成度を高くして提供します。

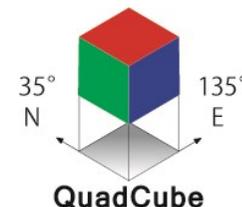
## 価格表

	(税別)
クアッドキューブ・デ変研QCライブラリの販売	
デ変研QCライブラリ for Linux 移動体キット海外含む	4,000,000円 / 初回のみ
デ変研QCライブラリ for Linux 移動体年間保守費海外含む	600,000円 / 年
デ変研QCライブラリ for Linux 移動体キット	2,000,000円 / 初回のみ
デ変研QCライブラリ for Linux 移動体年間保守費	300,000円 / 年
デ変研QCライブラリ for Linux 固定PCキット	240,000円 / 初回のみ
デ変研QCライブラリ for Linux 固定PC年間保守費	36,000円 / 年
デ変研QCライブラリ 複製使用权(1PC・台)	4,600円 / 1(PC・台)
デ変研QCライブラリ 教育サブスクリプション	80,000円 / 年
デ変研QCライブラリ 教育複製使用权(1PC・台)	無料 / 1(PC・台)
クアッドキューブの開発仕様書(複製使用权1PC・台付)	4,600円 / 1(PC・台)

※ 面積1m×1m以下の移動体用途の場合、固定PC版をお求めください。

© Copyright QuadCube Limited 2017. All rights reserved

# お問い合わせ先



ご清聴、ありがとうございました。

補足事項としては、Linux Business Initiative ( LBI )  
の会長しています。お知らせします。  
<http://www.lbi.gr.jp/>

## クアッドキューブ株式会社

E-mail [quadcube@dehenken.co.jp](mailto:quadcube@dehenken.co.jp)  
URL <http://www.derimo.net/>

TEL 03-6715-8404  
FAX 03-6715-8405

### 所在地

〒144-0052 (最寄り駅 JR蒲田駅、京急蒲田駅 徒歩6分)  
東京都大田区蒲田4-42-3-201 BIZ Kamata

