

# kinect と音響信号を用いた視覚障害者支援デバイスの性能向上

関西大学 環境都市工学部 建築学科  
建築環境工学第 I 研究室  
建 10-90 古川 晃司  
指導教官 豊田 政弘

## 目次

1. 序論 .....	2
1. 1 研究背景 .....	2
1. 2 kinect について .....	3
1. 3 既往研究 .....	4
2. 研究目的 .....	9
3. 問題点と改善案 .....	10
3. 1 問題点 1 .....	10
3. 1. 1 提案 1 .....	11
3. 1. 2 提案 2 .....	13
3. 2 問題点 2 .....	14
3. 2. 1 提案 .....	15
4. 評価 .....	16
4. 1 実験方法 .....	16
4. 2 結果 .....	17
4. 3 考察 .....	20
5. 総括 .....	21
5. 1 まとめ .....	21
5. 2 今後の課題 .....	22
参考文献 .....	23

## 1. 序論

### 1. 1 研究背景

今日、視覚障害者が自立して暮らせるように盲導犬が活躍し、点字ブロックが設置されている。これらは視覚障害者の方々の手助けとなっており、普段の生活において必要不可欠なものとなっているが、これらを使用すると常に片手は塞がった状態である。しかし、「音」で瞬時に物や人の位置が把握できれば、両手が空いた状態で活動することができ、これまで以上に行動の幅を広げることができる。

本研究では、上記実現のために kinect と音響信号を用いた視覚障害者支援デバイスの実用化に向けて、デバイスの性能向上を行う。

## 1. 2 kinect について



写真1 kinect

写真1がkinectである。kinectとは主にコントローラーを使わないゲームで使用される機器である。カメラ、マイクが内蔵されており、動きや音声を認識することができる。また、kinectには赤外線を用いた深度センサーが搭載されており、障害物との距離を測ることができる。

### 仕様

- ・解像度 640×480
- ・角度 横方向 約57° 縦方向 約43°
- ・距離計測範囲 約0.4～4.0m

### 1. 3 既往研究

以下に kinect を用いた視覚障害者向けの支援装置に関する既往研究を紹介する。

#### ◎コンスタンツ大学での kinect を用いた視覚障害者向けの補助装置[1][2]

写真2のように kinect が取り付けられたヘルメットで壁や障害物への距離を測定し、写真3の振動装置を腰に巻きつけて着用者に警告するというシステムである。また、壁やドアに貼られたシンボルマークを検出し、距離などを音声で伝えることも可能である。



写真2 Kinect を取り付けられたヘルメット

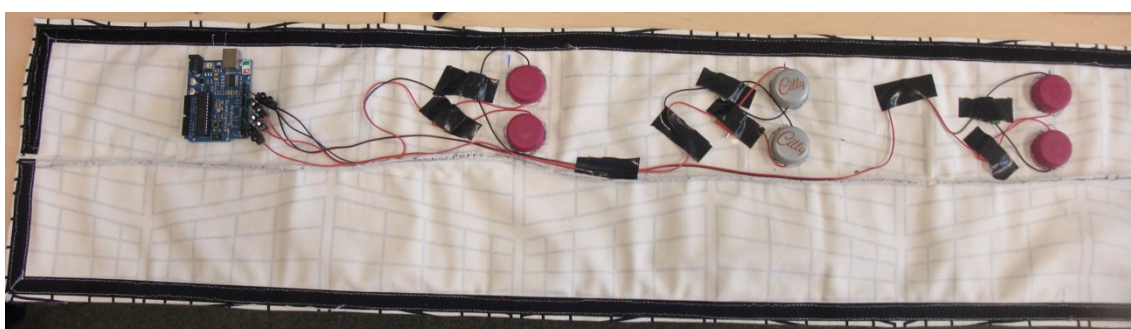


写真3 振動装置

◎久保の研究[3]

○目的

kinect と音響信号を用い、「音」を使って視覚障害者が物や人の位置を認識し、行動できるようになるためのデバイスを開発する。

○開発されたシステムの概要

kinect のドライバ、MATLAB、pure data、音の出力ドライバを用い、図 1、図 2 の範囲で障害物を検出し音を提示する。

$r = 0.4 \sim 4.0\text{m}$

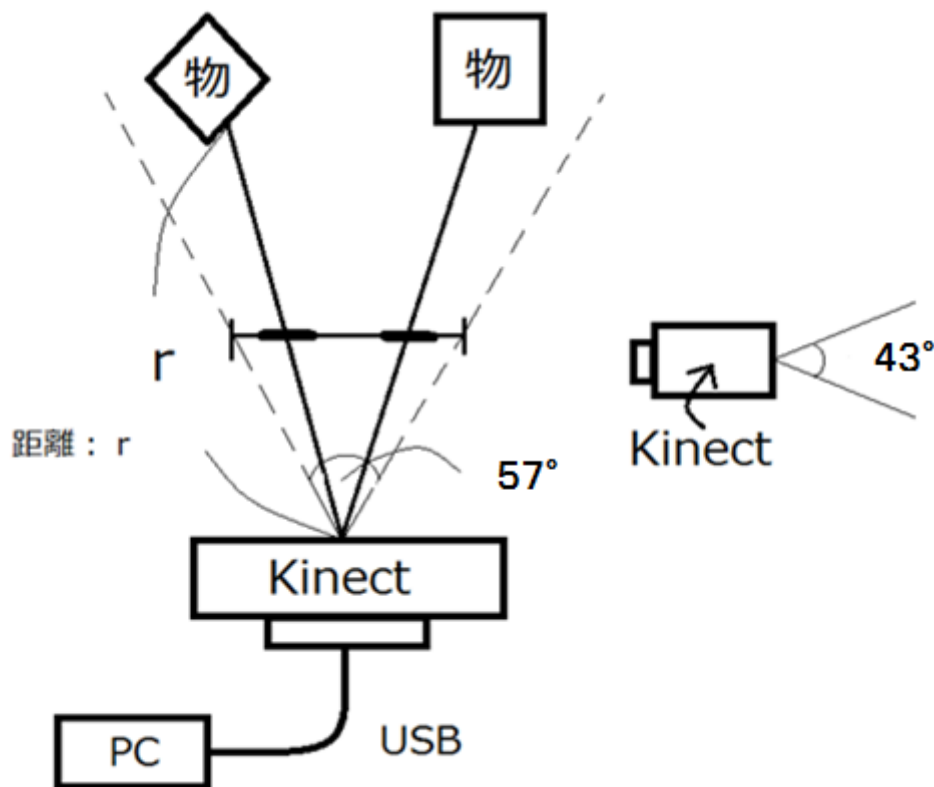


図 1 上からの図

図 2 横からの図

○デバイスの処理手順

- ・ kinect で 640×480 の深度データを取得
- ・ 8×6 のサブセットに分割
- ・ 最も深度の小さいサブセットの番号と深度を検出
- ・ 図3のサブセットの番号と深度に応じて音源を作成
- ・ 骨伝導ヘッドフォンで音を提示

0	6	12	18	24	30	36	42
1	7	13	19	25	31	37	43
2	8	14	20	26	32	38	44
3	9	15	21	27	33	39	45
4	10	16	22	28	34	40	46
5	11	17	23	29	35	41	47

図3 サブセット番号

○音の鳴り方

1030Hz 高	0	6	12	18	24	30	36	42
↑ ↓	1	7	13	19	25	31	37	43
	2	8	14	20	26	32	38	44
	3	9	15	21	27	33	39	45
	4	10	16	22	28	34	40	46
	5	11	17	23	29	35	41	47
980Hz 低	左 ←						→ 右	

図4 サブセット番号による提示音

・ 図4のようにサブセットの位置の方向からの HRTF を畳み込んだ音を提示する。

・ 音量は深度が大きくなるにつれて、小さくなる。

・ 距離によって提示する音のテンポが変わる。

近い (0.4~0.5m) ピピピピピ

中間 (0.6~0.7m) ピ ピ ピ

遠い (0.8m~) ピ ピ

・ 図4のようにサブセットの高さによって周波数を変化させる。

0 番の行 1030Hz

1 番の行 1020Hz

2 番の行 1010Hz

3 番の行 1000Hz

4 番の行 990Hz

5 番の行 980Hz



## ○結果

障害物の位置認識は、左右の判別は比較的容易で、正解率が高いが、上下方向の判別は難しく正解率はかなり低い結果となった。

また、初めて使う状態では難しいが、少しデバイスに慣れる練習を重ねると、ゆっくりと音が鳴る方向を確認しながら進むことで目隠しをし、デバイスを装着した状態で簡易な迷路を接触せずに抜けることも可能であった。

## 2. 研究目的

kinect と音響信号を用い、「音」を使って視覚障害者が物や人の位置を認識し、行動できるようになるためのデバイスの性能を向上させることを目的とする。そして、最終的にはより直感的に行動でき、また正確に障害物の位置を把握できるデバイスとなることを目的とする。

次節以降、久保の研究[3]で確認された問題点を挙げ、それに対する改善案を提案する。

### 3. 問題点と改善案

#### 3. 1 問題点 1

[2]の既往研究では上下の判別は難しいが、左右の判別は容易であるという結果が出た。しかし、窓やガラスの前では上下だけでなく、左右の判別も困難であった。

写真4に kinect を向けると、写真5～7のように表示される。色は深度を表わしており、深度が小さいほど青く、遠いほど赤く表示される。しかし、窓の一部分だけ色が変わっている部分がある。これは太陽光がガラスを透過した時の赤外線が kinect に入り込むのが原因だと考えられる。この部分は色を変えながら、点滅している。この点滅ノイズが発生している部分で誤った深度データを検出してしまうことが窓やガラスでは距離が正しく測ることができない原因であった。



写真4 Kinect を向けた窓

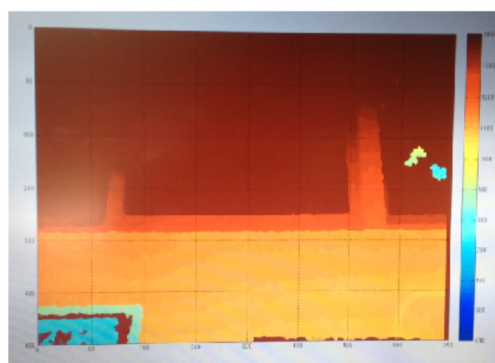


写真5 ノイズの様子

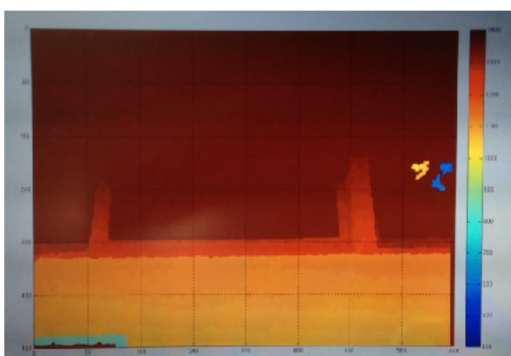


写真6 ノイズの様子

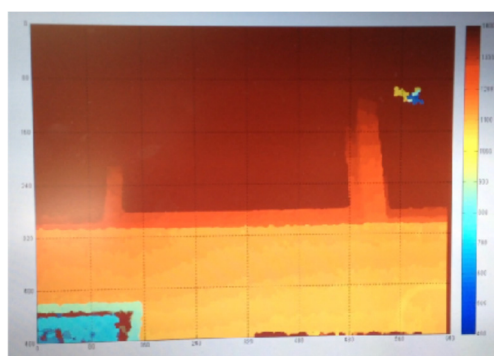


写真7 ノイズの様子

### 3. 1. 1 提案 1

常に色を変化させ点滅しているということは、ノイズは速い速度で動いているモノと誤認識されている。そこで、システム上感知する必要のない速度のモノを排除する。

装着者はもちろん徒歩での移動と想定する。目が見えない状態で、音が聞こえてから反応するということを想定すると、周辺の自転車や走っている人の速度に反応することはまだ不可能である。そこで、徒歩ですれ違うよりも速い速度のモノを排除する。

不動産情報で駅までの時間などで用いられている徒歩の速度 80m/分 を基準とすると、

徒歩 1 人	80m	1 分
徒歩 1 人	0.67m	0.5 秒
徒歩すれ違い	1.34m	0.5 秒

となる。以上より、0.5 秒に 1.34m 以上進むモノを排除する作業を組み込む。

提案 1 を加えるために、まず 1 秒間に深度データを取得する回数を 2fps から 20fps に変更する。すなわち、0.5 秒に 1 回データ取得から 0.05 秒に 1 回データ取得とし、音は 0.5 秒に 1 回提示する。従って、音を提示する時に 10 回分の深度データを取得している。この 10 回分のデータを基に徒歩ですれ違うよりも速い速度のモノを排除する。0.5 秒間の 10 回分のデータの最大の深度データと最小の深度データを比較し、1.34m 以上の差があれば深度を 10m に強制的に書き換えることで検出対象から排除する。

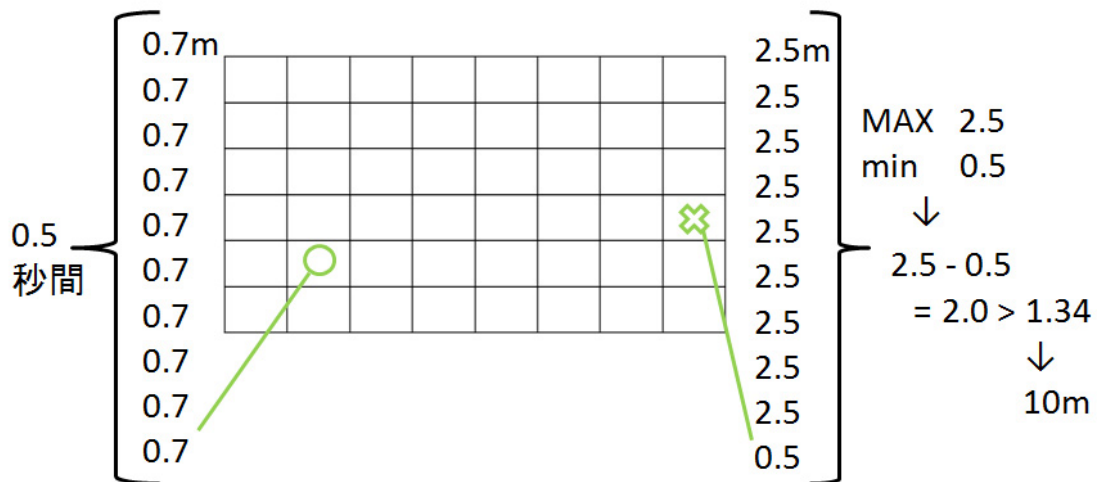


図5 深度データ取得例

例えば、図5の左のサブセット（○印位置）では1～10回目のすべてで0.7mのデータを取得したとする。右のサブセット（×印位置）では1～9回目は2.5mで10回目のみ0.5mのデータを取得したとする。

従来のシステムでは、右のサブセットの中で10回目のデータのみが1～9回目と比べ大きく変化しており、誤ったデータである可能性があるが、音を鳴らす10回目のデータで最小の深度0.5mである右のサブセットから音を提示してしまう。

今回の提案では、左のサブセットでは最大最小深度共に0.7mであるので、10回目のデータ0.7mで作業を進める。右のサブセットでは最大深度が2.5m、最小深度が0.5mである。これらの差は2.0mであり1.34mという基準よりも大きいので、0.5mを10mに書き換える。そうすることによってノイズと思われる右のサブセットの値を排除でき、左の0.7mを検出し音を提示することができる。

### 3. 1. 2 提案2

問題点1の点滅ノイズはたいていの場合窓ガラスの中の様々な場所で起こるのではなく、ある1つの場所で起こっている。また、そのノイズが起こる決まった場所の範囲はあまり大きくなく、1つのサブセット内に収まることが多い。そこであるサブセットの深度と上下左右斜めのサブセットの深度8つと比較し、周囲のすべてのサブセットと2.0m以上差がある場合はノイズであると判断し、10mに書き換えるという作業を加える。例として図6のような深度データが得られた場合には中央のサブセットの深度は10.0mに書き換えられる。

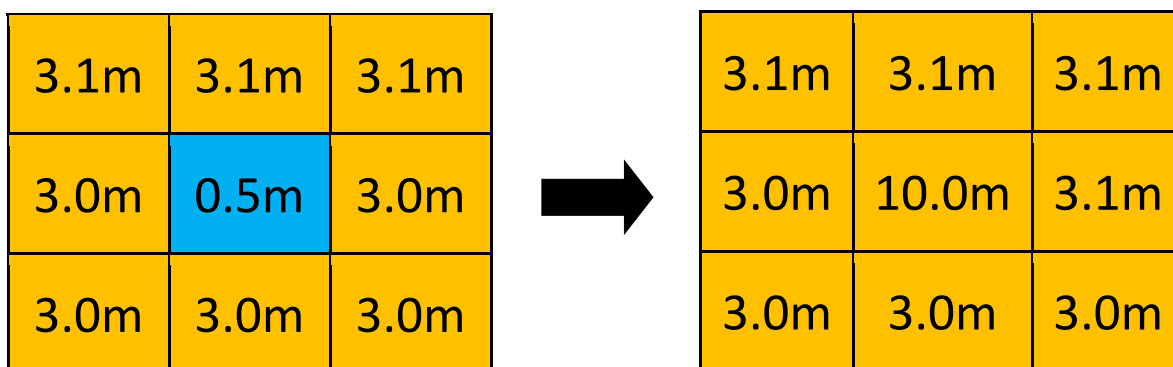
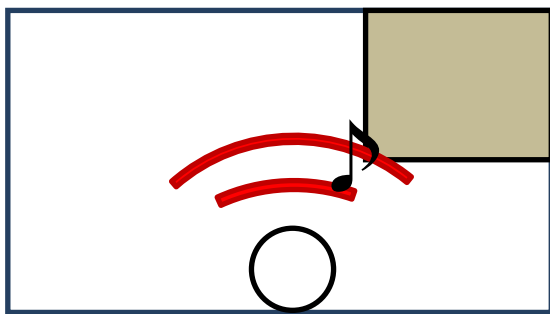


図6 サブセットの一部

### 3. 2 問題点2

左右に障害物がある場合、直進可能か判断し辛い。障害物の正確な位置を判断する場合、体の向きを変えた時の音が鳴る方向で判断する。図7のように右のみに障害物がある場合、左を向くと途中で音が鳴りやみ、右側に障害物があったとはっきりわかる。図8のように左右に障害物がある場合左を向いても常に音が鳴り続け、正面方向が空いていても壁だと思いつま可能性はある。



↓ 左を向くと

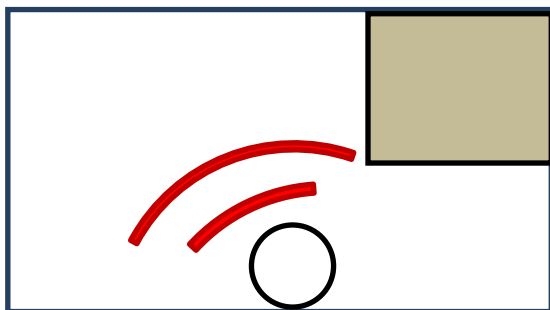
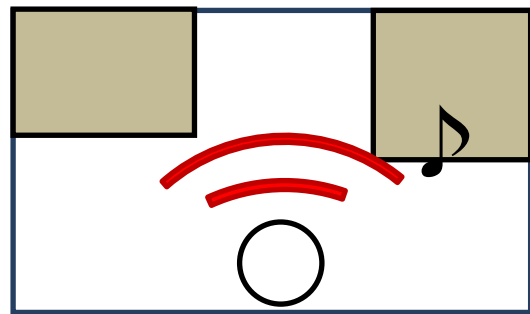


図7 右のみに障害物



↓ 左を向いても

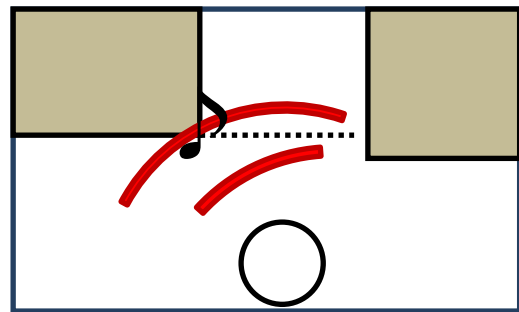


図8 左右に障害物

### 3. 2. 1 提案

障害物の左右の判別は容易であるが、正確な障害物の方向は認識し辛い。そこで新たに、障害物が正面にあるかを判断するシステムを作る。すなわち、直進可能かを判断するシステムを作る。

mrmr というアプリを使い iPhone とデバイスを連動させる。そして、iPhone の画面を触った時に、最も深度の小さい障害物が図9の18、24の列の場合、直進不可と判断し、何も起こらない。0、6、12、30、36、42の列の場合、直進可能と判断し、音を提示する。この音は障害物の深度による音量の変化はなく、HRTF も畳み込んでいない音で障害物の位置を示す音とは全く別の音である。

0	6	12	18	24	30	36	42
1	7	13	19	25	31	37	43
2	8	14	20	26	32	38	44
3	9	15	21	27	33	39	45
4	10	16	22	28	34	40	46
5	11	17	23	29	35	41	47

図9 サブセット



## 4. 評価

### 4. 1 実験方法

問題点 1 に対する改善策について物理評価実験を行い、その有効性を検証する。写真 8 のように kinect を窓に向けた状態で固定する。窓と kinect の間に障害物を置き、常に決まったサブセットから音が鳴る状態を作る。今回の実験では、下から 1 行目、左から 2 列目のサブセットで行う。そして、1 分間デバイスを動かし、いずれかのサブセットを検出した全回数と、下から 1 行目、左から 2 列目のサブセットを検出した回数をカウントする。

この実験を 1 時間ごとに行う。「3. 1. 1 提案 1」「3. 1. 2 提案 2」を組み込む前と組み込んだ後の MATLAB プログラムで実験を行い、時刻による違い、点滅ノイズによるエラーが起こる割合等を比較する。



写真 8 実験の様子

## 4. 2 結果

$$\frac{\text{下から 1 行目、左から 2 列目のサブセットを検出した回数}}{\text{いずれかのサブセットを検出した全回数}} \times 100 \quad [\%]$$

で正しく検出された割合を算出し、比較する。

表 1 に正しく検出された割合の結果をまとめる。

	11～14 時	15 時	16 時	17 時	18 時
改善前	100%	52.0%	34.3%	43.1%	100%
改善後	100%	89.0%	73.4%	91.8%	100%

表 1

(小数点第二位を四捨五入して表示)

11 時～14 時では、図 1 0 のように全く点滅ノイズは見られず、エラーも全く起こらなかった。

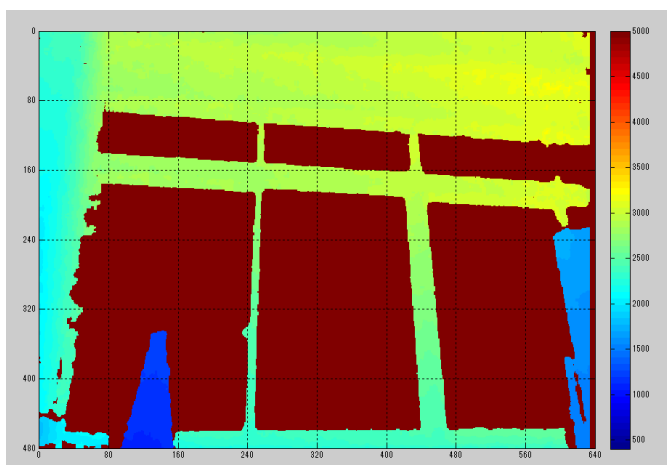


図 1 0 11～14 時の様子

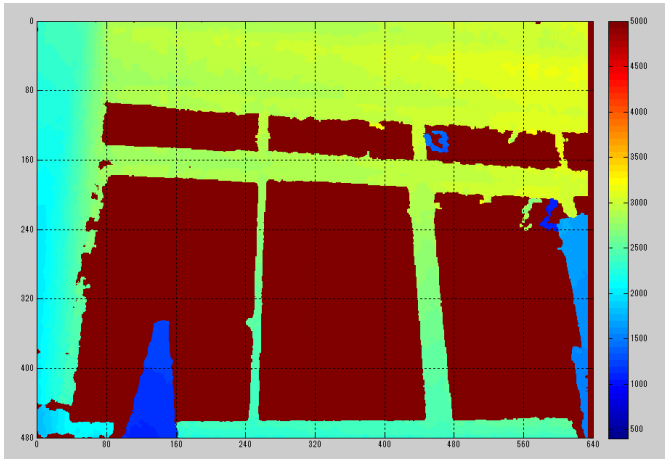


図 1 1 15 時の様子

図 1 1、図 1 2、図 1 3 のように 15~17 時は多くの点滅ノイズが発生しており、16 時に最も多くのノイズが見られた。

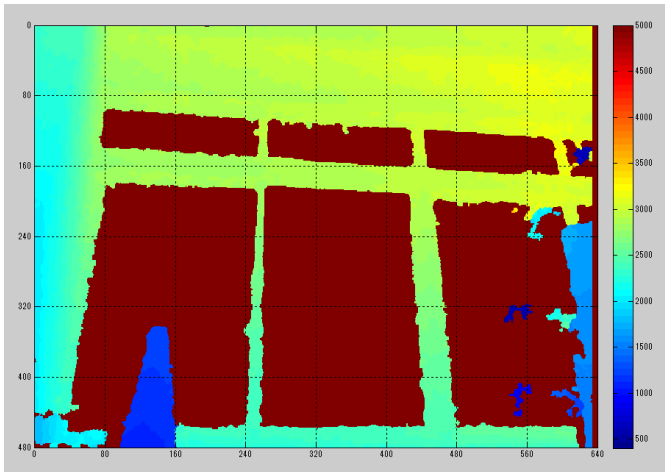


図 1 2 16 時の様子

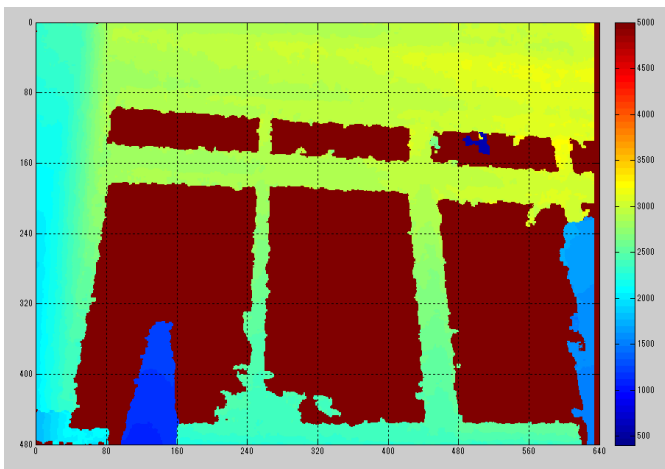


図 1 3 17 時の様子

18 時には、図 1 4 のように窓ガラスの一部がサッカーグラウンドを照らすライトの影響で緑色に変化しているが、エラーを起こすような点滅ノイズは確認されなかった。

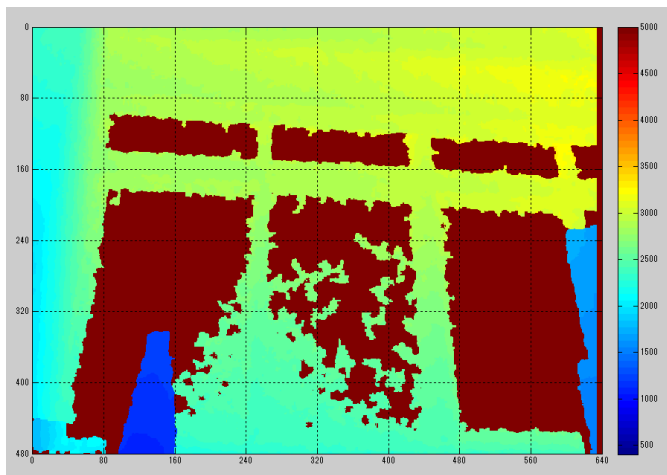


図 1 4 18 時の様子

#### 4. 3 考察

表 1 より全ての時間帯で改善後の方がより正しく障害物のあるサブセットを検出する事ができた。つまり提案 1、提案 2 は点滅ノイズによるエラーに対して大変効果があると判断できる。

14 時まではノイズは発生せず、15 時以降にのみ急激に発生した。日没前の時間帯にのみ多くの点滅ノイズが発生すると考えられる。

また、18 時の実験時にはすでに日が沈んでいた。図 15 のように屋外のライトの影響で一部色に変化しているが、特にノイズは起こらなかった。つまり点滅ノイズはライトでは起こらず、太陽光でのみ引き起こされる現象だと考えられる。

## 5. 総括

### 5. 1 まとめ

kinect と音響信号を用いた視覚障害者支援デバイスを実用化させていくには、より直感的に反応できるようにすること、より障害物認識精度の向上させることが求められており、まだまだ改良すべき点が多々ある。

本研究では窓ガラスでの点滅ノイズを大幅に減少させることができ、正面方向の障害物を確認する新たなシステムを組み込んだ。これにより既存システムに比べ大幅に誤作動を減らし、障害物の位置認識の精度を向上させることができた。

## 5. 2 今後の課題

点滅ノイズによるエラーは大幅に減少させることができたが、すべてを取り除くことはできなかった。徒歩ですれ違うよりも遅い速度のモノを排除することは、実際にデバイスを装着し、外に出た時に周囲の人を検出できなくなってしまうので、システム上望ましくない。従って、さらにノイズを取り除くためには新たな手段の考案が必要である。今回、点滅ノイズを排除するために速度制限を設けたが、速度が速いモノが装着者に向かってきている場合は警告音を出すなどのシステムを検討すべきである。

また、kinect の機器の設計上、足元の障害物を認識することができない。そこで、kinect を少し下に向け、足元用の距離認識システムを新たに構築するなどの対策も今後必要である。

## 参考文献

[1]

GameSpark

URL(<http://www.gamespark.jp/article/2011/03/18/27138.html>)

[2]

Human-Computer Interaction

URL(<http://hci.uni-konstanz.de/blog/2011/03/15/navi/?lang=en>)

[3]

久保卓真, kinect と音響信号を用いた視覚障害者支援デバイスの開発  
関西大学卒業論文, 2013 年



## MATLAB

```
% get depth image via Kinect

close all;
clear all;
addpath('./Mex');

% create context with xml file
context = mxNiCreateContext('Config/SamplesIR_DepthConfig.xml');

% numbers of grid
nx = 8;
ny = 6;

% initialise figure
depth_width = 640;
depth_height = 480;
figure, h1 = imagesc(zeros(depth_height, depth_width, 'uint16'));
axis equal;
set(gca, 'XTick', [0 80 160 240 320 400 480 560 640], 'XLim', [0 depth_width]);
set(gca, 'YTick', [0 80 160 240 320 400 480], 'YLim', [0 depth_height]);
set(gca, 'Clim', [400 5000]);
colorbar;
grid on;
hold on;

% osc address setting
osc_add = osc_new_address('127.0.0.1', 3334);

% pixels per grid
px = depth_width / nx;
py = depth_height / ny;

% depth matrix for grid discretization
```

```

a      = zeros(ny, nx, 10, 'uint16');
atmp2  = zeros(ny, nx, 'uint16');

count = 0;

% loop
while 1;

    count = count + 1;

    tic

    % capture depth image
    mxNiUpdateContext(context);
    [ir, depth] = mxNiIRImage(context);

    % remove zero
    for ii = 1 : depth_width
        for jj = 1 : depth_height
            if depth(jj, ii) == 0
                depth(jj, ii) = 10000;
            end
        end
    end

    % search minimum value in each cell
    for ii = 1 : nx
        for jj = 1 : ny
            a(ny-jj+1, nx-ii+1, count) = min(min(depth(py*(jj-1)+1 : py*jj,
px*(ii-1)+1 : px*ii)));
        end
    end

    if count == 10;
        for ii = 1: nx
            for jj = 1:ny

```

```

    for kk = 1:10
        atmp(kk) = a(ny-jj+1, nx-ii+1, kk);
    end
    if max(atmp)-min(atmp) > 1340;
        a(ny-jj+1, nx-ii+1, 10) = 10000;
    end;
    atmp2(ny-jj+1, nx-ii+1) = a(ny-jj+1, nx-ii+1, 10);
end;
end;
end;

% reject noise challenge
for ii = 2 : 7
    for jj = 2 : 5
        if atmp2(jj, ii)-atmp2(jj+1, ii+1) > 2000 &...
            atmp2(jj, ii)-atmp2(jj+1, ii) > 2000 &...
            atmp2(jj, ii)-atmp2(jj+1, ii-1) > 2000 &...
            atmp2(jj, ii)-atmp2(jj, ii+1) > 2000 &...
            atmp2(jj, ii)-atmp2(jj, ii-1) > 2000 &...
            atmp2(jj, ii)-atmp2(jj-1, ii+1) > 2000 &...
            atmp2(jj, ii)-atmp2(jj-1, ii) > 2000 &...
            atmp2(jj, ii)-atmp2(jj-1, ii-1) > 2000
            atmp2(jj, ii) = 10000;
        end
    end
end
end

for ii = 1
    for jj = 2 : 5
        if atmp2(jj, ii)-atmp2(jj+1, ii+1) > 2000 &...
            atmp2(jj, ii)-atmp2(jj+1, ii) > 2000 &...
            atmp2(jj, ii)-atmp2(jj, ii+1) > 2000 &...
            atmp2(jj, ii)-atmp2(jj-1, ii+1) > 2000 &...
            atmp2(jj, ii)-atmp2(jj-1, ii) > 2000
            atmp2(jj, ii) = 10000;
        end
    end
end

```

```

    end
end

for ii = 8
    for jj = 2 : 5
        if atmp2(jj, ii)-atmp2(jj+1, ii) > 2000 & ...
            atmp2(jj, ii)-atmp2(jj+1, ii-1) > 2000 & ...
            atmp2(jj, ii)-atmp2(jj, ii-1) > 2000 & ...
            atmp2(jj, ii)-atmp2(jj-1, ii) > 2000 & ...
            atmp2(jj, ii)-atmp2(jj-1, ii-1) > 2000
                atmp2(jj, ii) = 10000;
            end
        end
    end
end

for ii = 2 : 7
    for jj = 1
        if atmp2(jj, ii)-atmp2(jj+1, ii+1) > 2000 & ...
            atmp2(jj, ii)-atmp2(jj+1, ii) > 2000 & ...
            atmp2(jj, ii)-atmp2(jj+1, ii-1) > 2000 & ...
            atmp2(jj, ii)-atmp2(jj, ii+1) > 2000 & ...
            atmp2(jj, ii)-atmp2(jj, ii-1) > 2000
                atmp2(jj, ii) = 10000;
            end
        end
    end
end

for ii = 2 : 7
    for jj = 6
        if atmp2(jj, ii)-atmp2(jj, ii+1) > 2000 & ...
            atmp2(jj, ii)-atmp2(jj, ii-1) > 2000 & ...
            atmp2(jj, ii)-atmp2(jj-1, ii+1) > 2000 & ...
            atmp2(jj, ii)-atmp2(jj-1, ii) > 2000 & ...
            atmp2(jj, ii)-atmp2(jj-1, ii-1) > 2000
                atmp2(jj, ii) = 10000;
            end
        end
    end
end

```

```

    end
end

for ii = 1
    for jj = 1
        if atmp2(jj, ii)-atmp2(jj+1, ii+1) > 2000 & ...
            atmp2(jj, ii)-atmp2(jj+1, ii) > 2000 & ...
            atmp2(jj, ii)-atmp2(jj, ii+1) > 2000
                atmp2(jj, ii) = 10000;
            end
        end
    end
end

for ii = 8
    for jj = 1
        if atmp2(jj, ii)-atmp2(jj+1, ii) > 2000 & ...
            atmp2(jj, ii)-atmp2(jj+1, ii-1) > 2000 & ...
            atmp2(jj, ii)-atmp2(jj, ii-1) > 2000
                atmp2(jj, ii) = 10000;
            end
        end
    end
end

for ii = 1
    for jj = 6
        if atmp2(jj, ii)-atmp2(jj, ii+1) > 2000 & ...
            atmp2(jj, ii)-atmp2(jj-1, ii+1) > 2000 & ...
            atmp2(jj, ii)-atmp2(jj-1, ii) > 2000
                atmp2(jj, ii) = 10000;
            end
        end
    end
end

for ii = 8
    for jj = 6

```

```

        if atmp2(jj, ii)-atmp2(jj, ii-1) > 2000 & ...
            atmp2(jj, ii)-atmp2(jj-1, ii) > 2000 & ...
            atmp2(jj, ii)-atmp2(jj-1, ii-1) > 2000
                atmp2(jj, ii) = 10000;
            end
        end
    end
end

% extract the closest point
[b, ib] = min(atmp2);
[c1, ic] = min(b);
idx(1) = ib(ic);
idx(2) = ic;

% update figure
% if count == 10;
    set(h1, 'CData', fliplr(depth));
% end;
% display the closest cell
%for ii = 1 : nx - 1
%    plot([px*ii px*ii], [0 depth_height], 'Color', [0.3 0.3 0.3]);
%end
%for jj = 1 : ny - 1
%    plot([0 depth_width], [py*jj py*jj], 'Color', [0.3 0.3 0.3]);
%end
%if c1 < 1000
%    plot([px*(idx(2)-1) px*idx(2)], [py*(ny-idx(1)) py*(ny-idx(1))], 'w');
%    plot([px*(idx(2)-1) px*idx(2)], [py*(ny-idx(1)+1) py*(ny-idx(1)+1)], 'w');
%    plot([px*(idx(2)-1) px*(idx(2)-1)], [py*(ny-idx(1)) py*(ny-idx(1)+1)], 'w');
%    plot([px*idx(2) px*idx(2)], [py*(ny-idx(1)) py*(ny-idx(1)+1)],
'w');
%end

% draw
drawnow;

```

```

if count == 10;
% send osc message
m = struct( ...
    'path', '/depth', ...
    'tt', 'iif', ...
    'data', {[idx(1), idx(2), double(c1) / 1000]} ...
    );
osc_send(osc_add, m);
count = 0;
end;

1/toc

% pause
%pause(0.5);

end

% delete the context object
mxNiDeleteContext(context);

```

# Pure data

