

ビギナーズセミナー資料

# ゲーム周辺機器を使用した音響信号処理プログラミング ～Kinect, PS Eyeなど～

鎌土 記良

Noriyoshi Kamado

 Nara Institute of Science and Technology

# 本講演の目標

- 音響信号処理の知識を実際に試してみたい（収音系に限る）



矢張り音は聞いてみてなんぼ

- 初学者の心の中

- ⇒ 実装を研究に活かしたいが、プログラムを勉強している時間が…
- ⇒ 実装にかけるお金が… (DSPとか, ADとか, お金かからない?)

- そんな人がやる気が出るような夢のある情報を投げる

# 手軽に使える学習教材 Kinect, PS Eye

## ■ 小型カメラと 4ch マイクロホンアレーの複合装置

⇒ Kinect は更にレーザーレンジファインダー付き

(人体の高精度トラッキング可能)



PS Eye

¥2,380～



Kinect

¥11,181～

■ とにかく安い → 個人で趣味で買える値段

■ PC からUSBオーディオデバイスとして使える(ASIO可) →

PCのスペック次第で高度な信号処理もバリバリ実装可能

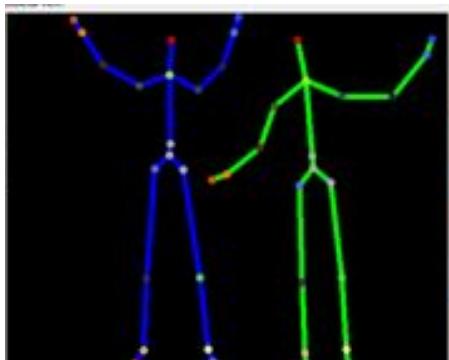


# Kinect™ のスペック

## ■ Kinect™

- Microsoft 社が同社 Xbox 360® 向けに開発したマルチモーダル・インターフェース
- 4チャネルマイクロホンアレー, RGB カメラ, 深度センサ

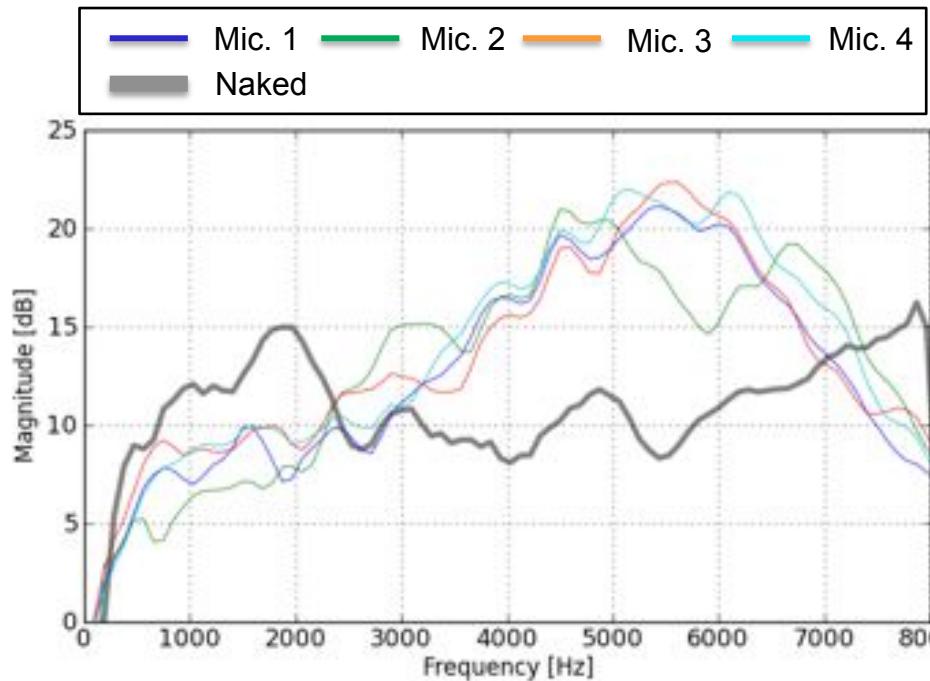
## ■ ポイントラッキング・スペック



- 話者頭部位置をミリメートル単位で取得可能
- Kinectを中心としたユークリッド座標系

## ■ 周波数特性

- ⇒ 素子間に 5 dB 程度特性差
- ⇒ 筐体により 3 kHz から 7 kHz を押し上げ
- ⇒ 1 kHz から 2 kHz の特性が抑圧



マイクロホンアレーの周波数特性



# Kinect™ のスペック

## ■ Kinect™

- Microsoft 社が同社 Xbox 360® 向けに開発したマルチモーダル・インターフェース
- 4 チャンネルマイクロホンアレー, RGB カメラ, 深度センサ

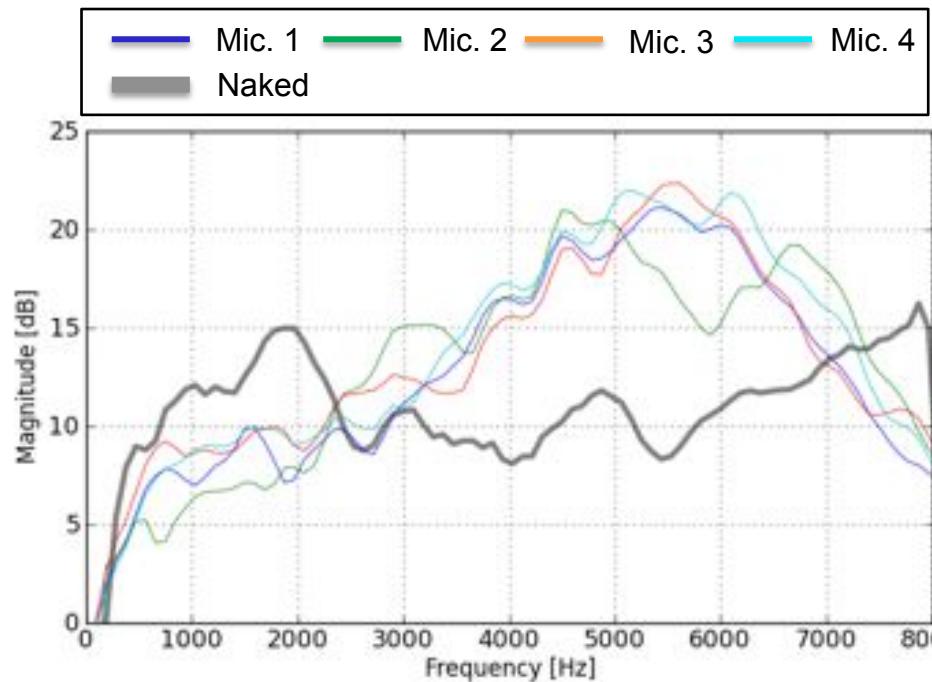
## ■ 内部構造



サンプリング周波数	16 [kHz]
量子化ビット数	16 [bit]
チャネル数	4

## ■ 周波数特性

- ⇒ 素子間に 5 dB 程度特性差
- ⇒ 筐体により 3 kHz から 7 kHz を押し上げ
- ⇒ 1 kHz から 2 kHz の特性が抑圧



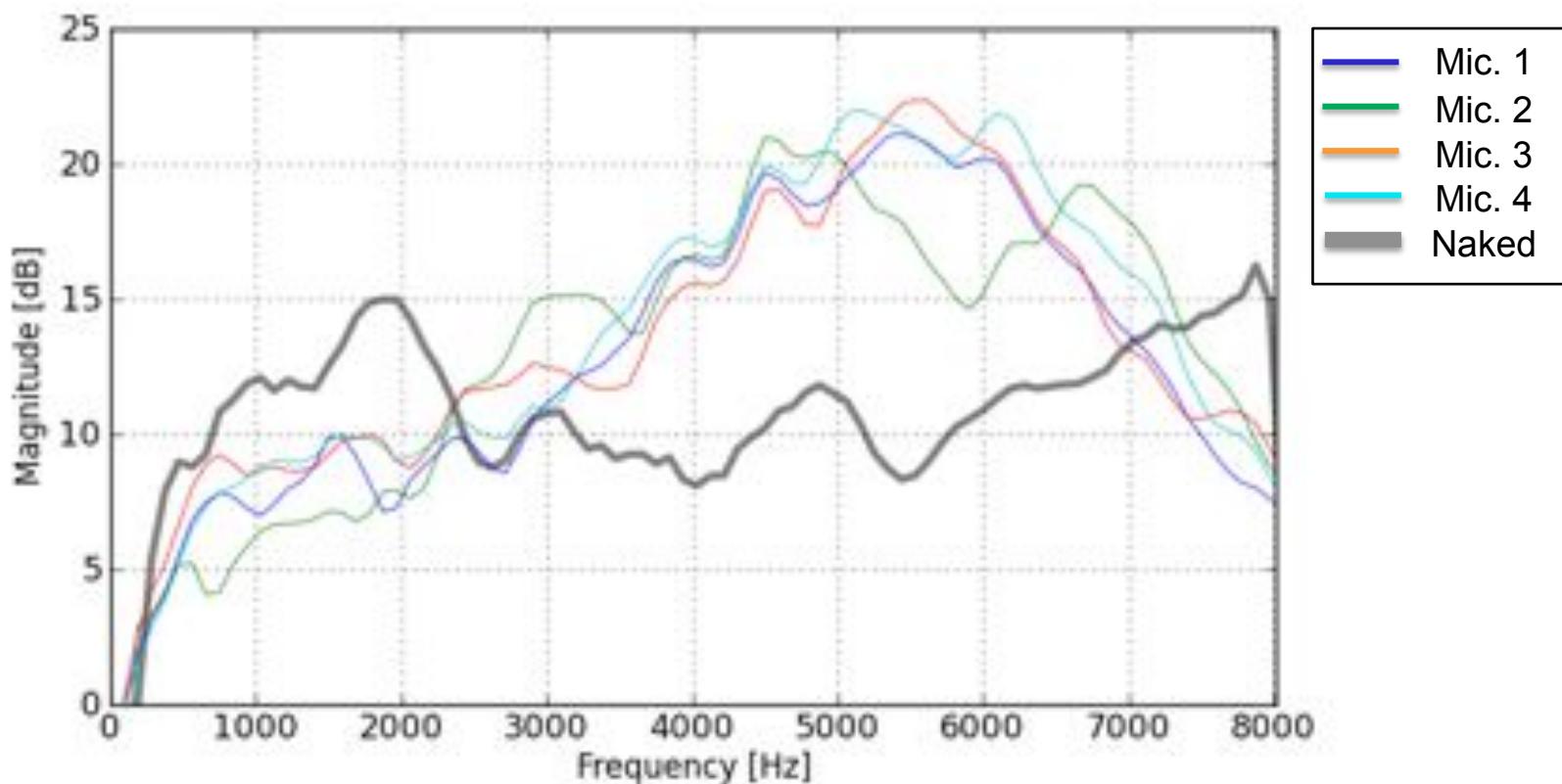
マイクロホンアレーの周波数特性



# Kinect のマイクロホンアレースペック

## ■ Kinect マイクロホン素子の周波数特性

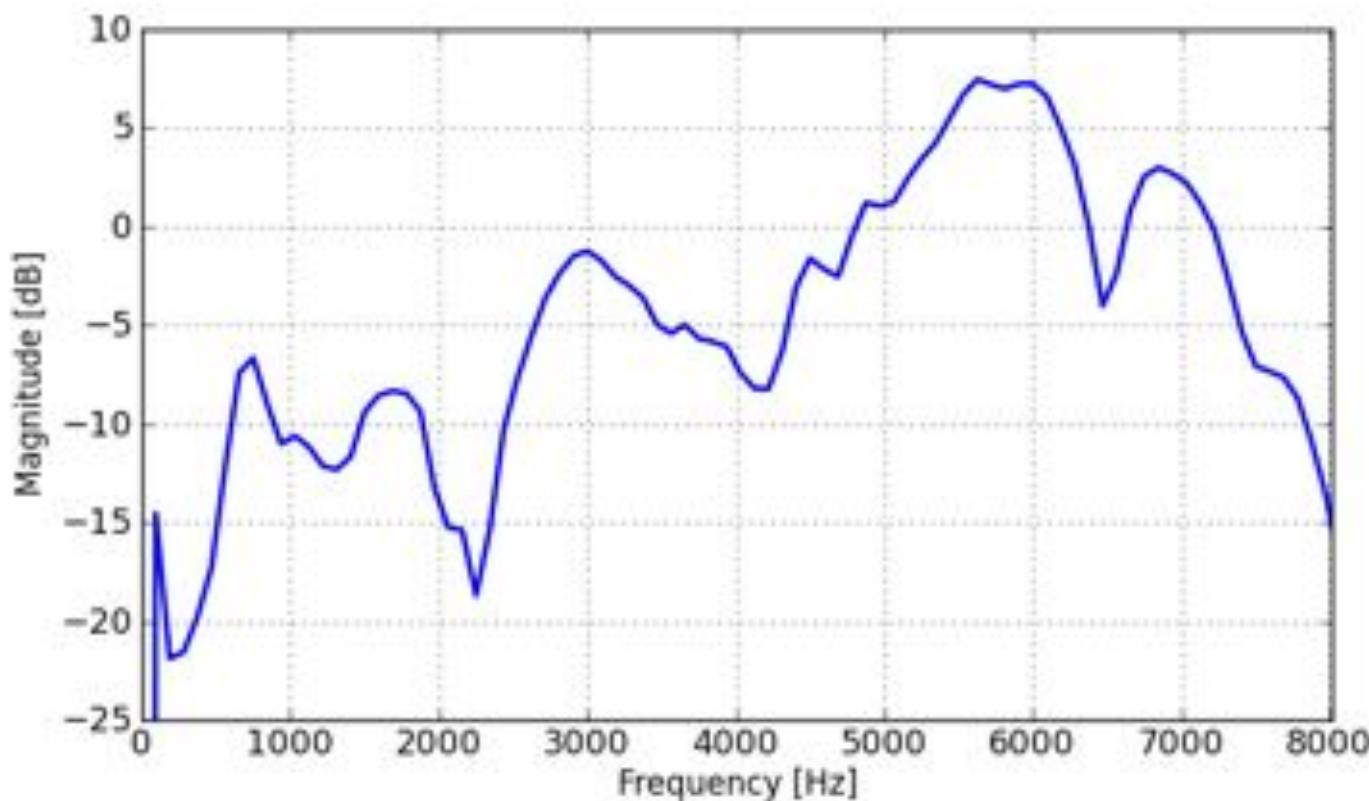
- ⇒ 素子間に 5 dB 程度特性差
- ⇒ 筐体により 3 kHz から 7 kHz が押し上げられる
- ⇒ 1 kHz から 2 kHz の特性が抑圧



# Kinect のマイクロホンアレースペック

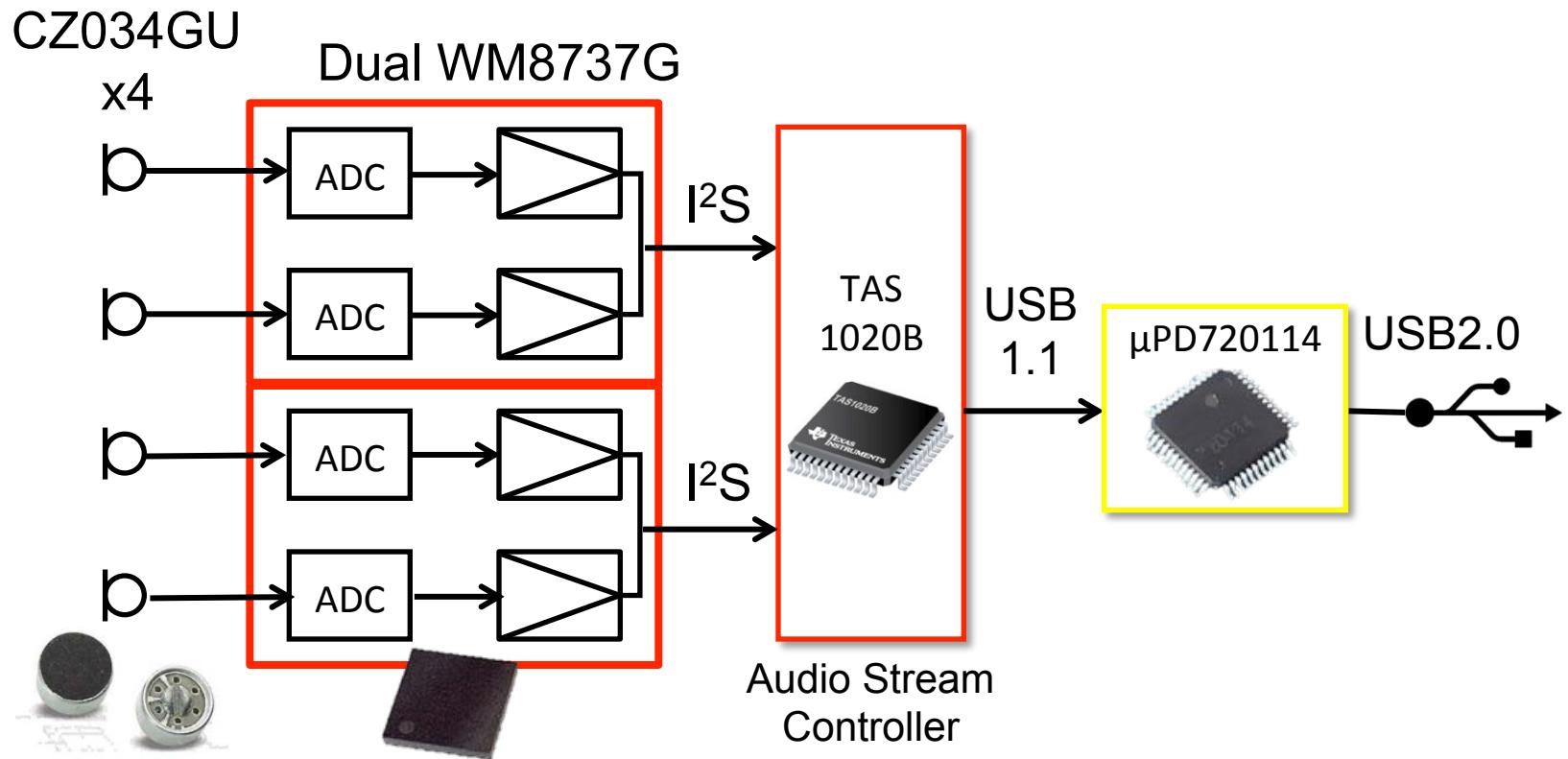
## ■ Kinect マイクロホン素子の周波数特性

- ⇒ 素子間に 5 dB 程度特性差
- ⇒ 筐体により 3 kHz から 7 kHz が押し上げられる
- ⇒ 1 kHz から 2 kHz の特性が抑圧



# Kinect のマイクロホンアレースペック

## ■ 要するに、ただの USB オーディオ



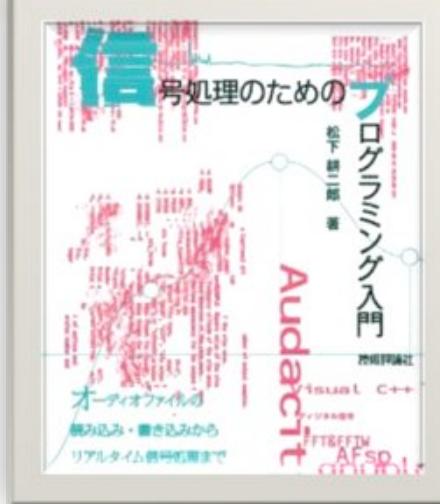
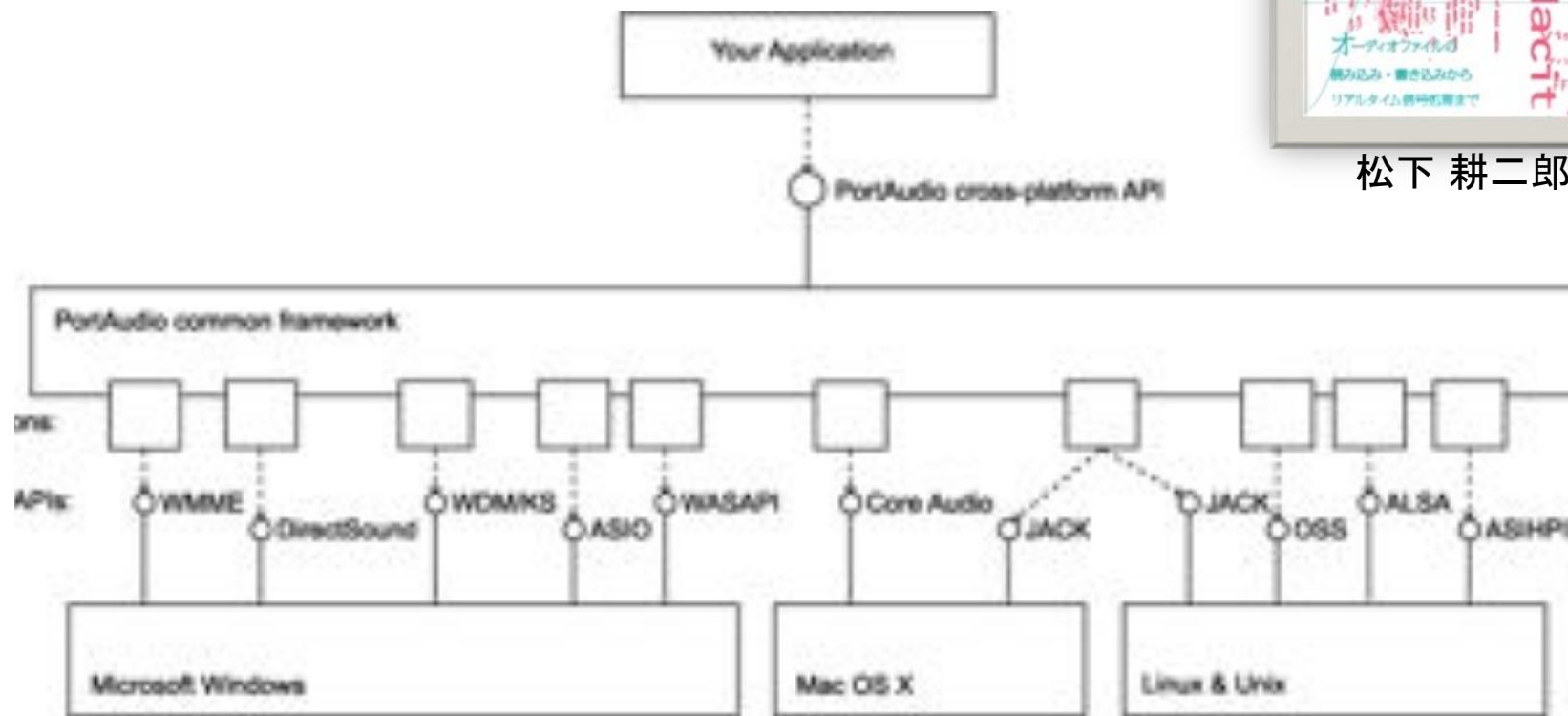
<http://www.ifixit.com/Teardown/Microsoft-Kinect-Teardown/4066/2>

# どうやって使うか？

オススメ！

## ■ ライ브ライアリを使えばクイックに開発可能

⇒ オススメは Port audio, libsndfile, fftw3, python



松下 耕二郎 著

# どうやって使うか？(Windows では)

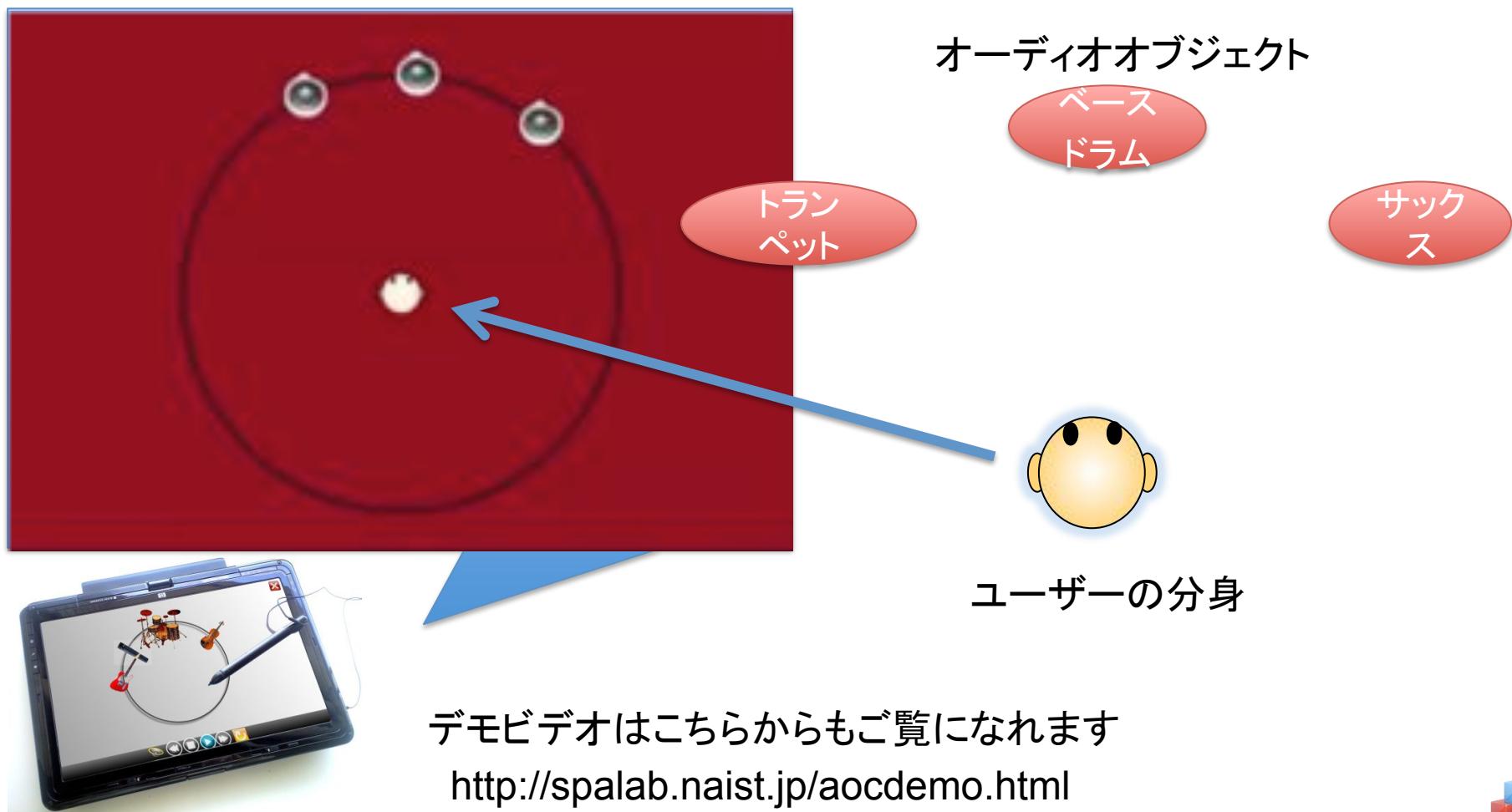
- ライブラリを使えばクイックに開発可能  
⇒ オススメは Port audio, libsndfile, fftw3, python など
- マイクロホン4つじゃ足りないよ！  
⇒ ASIO4ALL で複数の Kinect を一つのオーディオデバイスとして使用



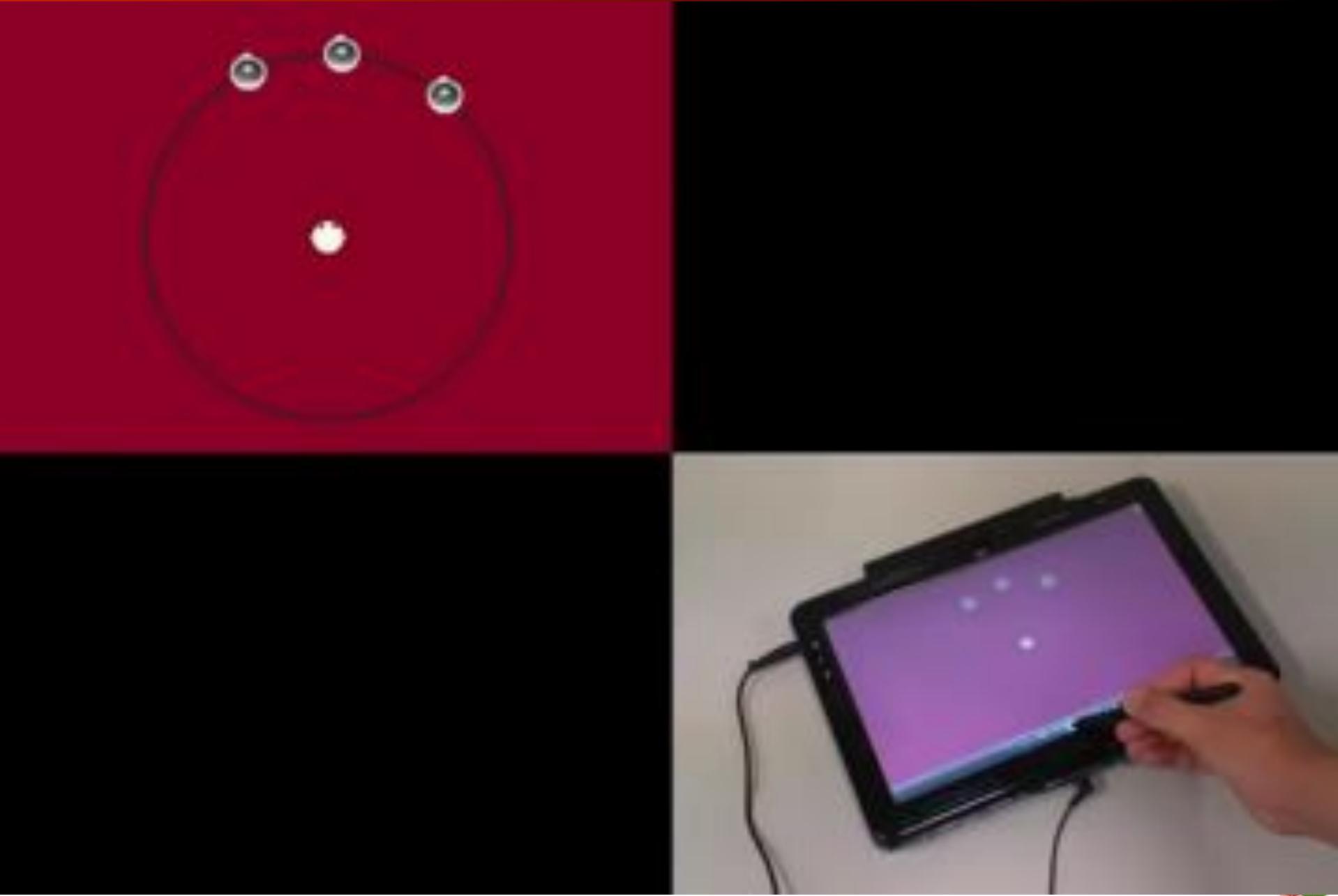
<http://wwwasio4all.com/>

# ライブラリを使用したアプリの一例 [Kamado et al., IWAENC 2010]

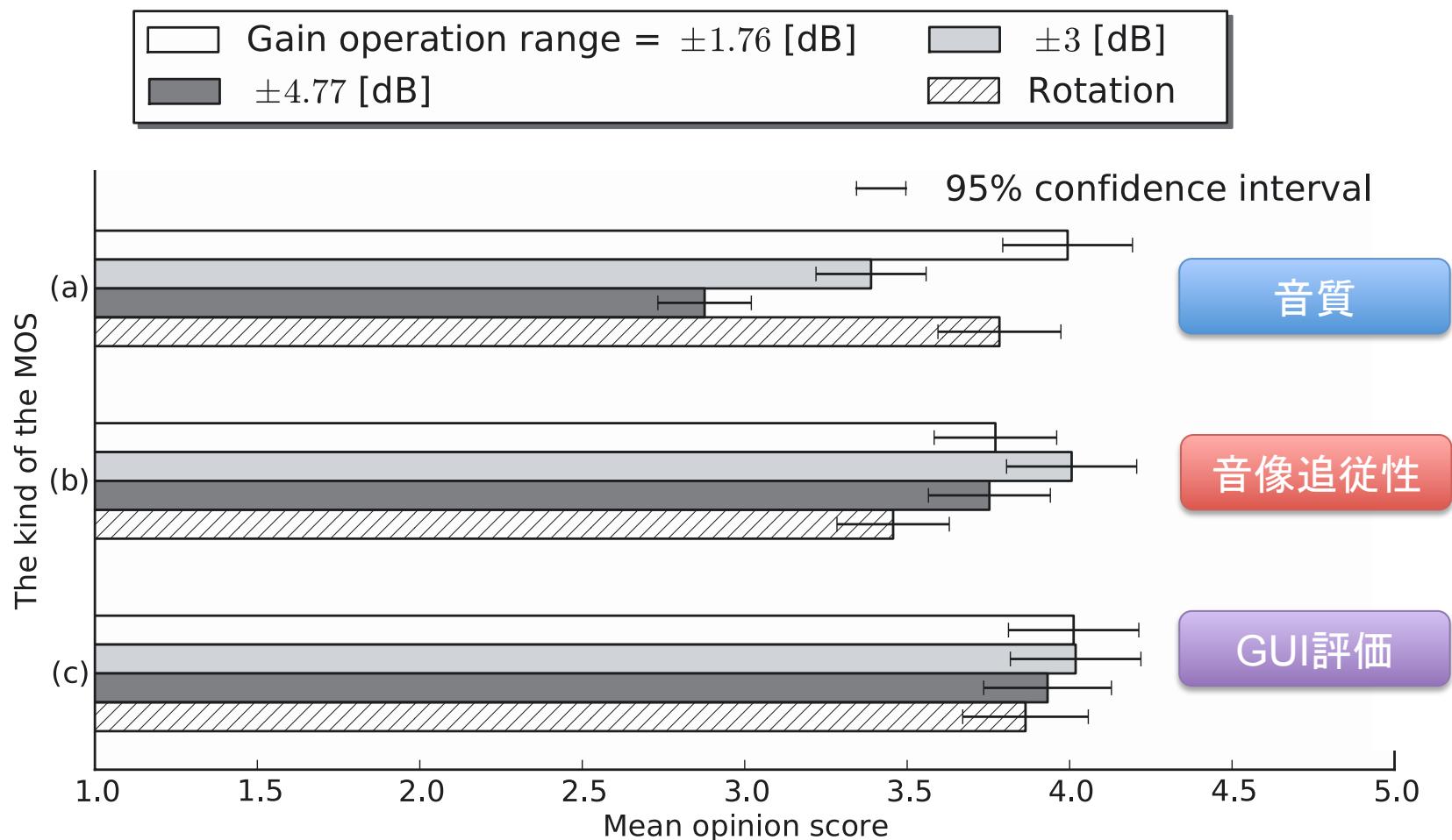
- 同一方位の楽器群のスペクトル情報を  $k$ -means クラスタリングで分離
- 分離した音源に対してインタラクティブな定位操作を可能に



## ライブラリを使用したアプリの一例 [Kamado et al., IWAENC 2010]



# ライブラリを使用したアプリの一例 [Kamado et al., IWAENC 2010]



- 過度の音量操作を除いて音質の劣化は少ない
- 音像の追従性も良い
- GUIの操作性も高い

# ライブラリを使用したアプリの一例

- Construct the 25ch linear loudspeaker array system
- Evaluate the wavefront of conventional (MCWS) and our proposed method in actual environment
- Primary source is located at  $(x,y,z)=(0.0,-1.0,1.22)$  m

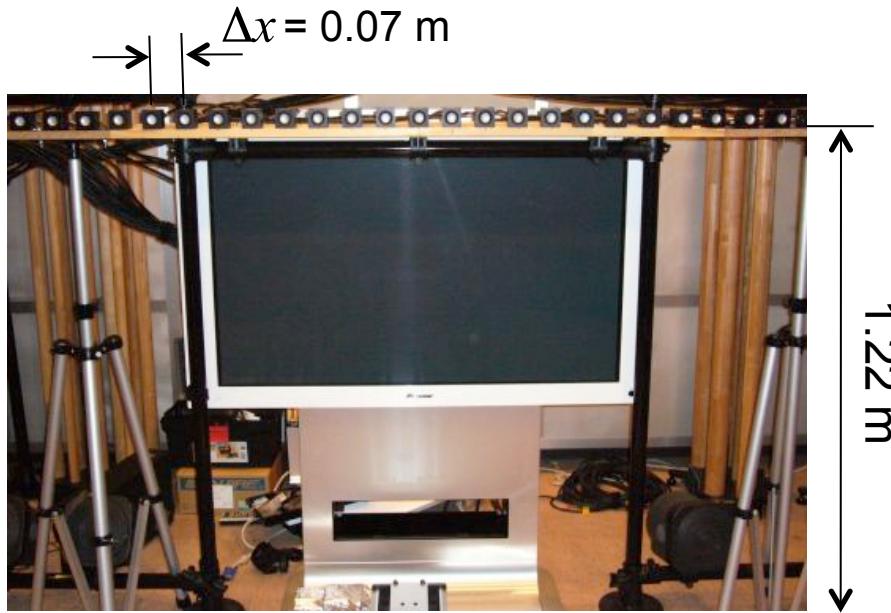
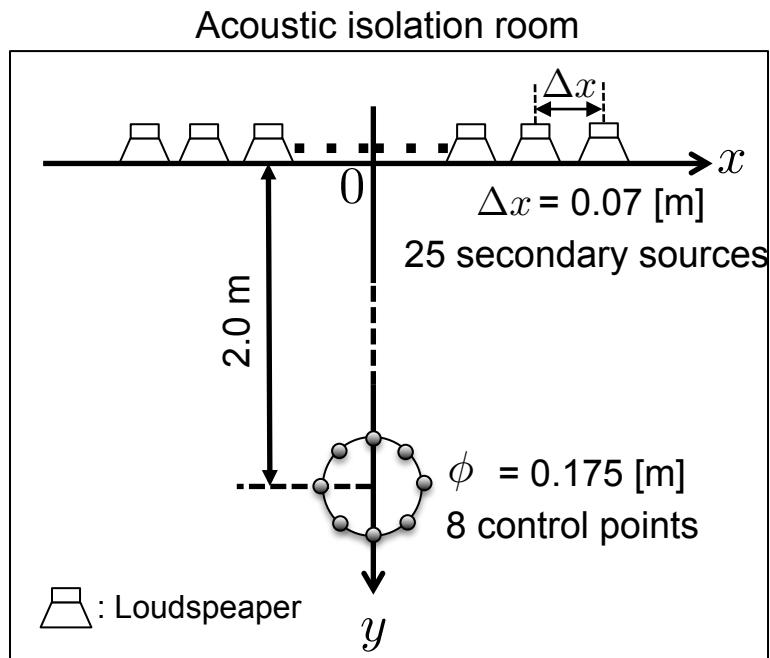


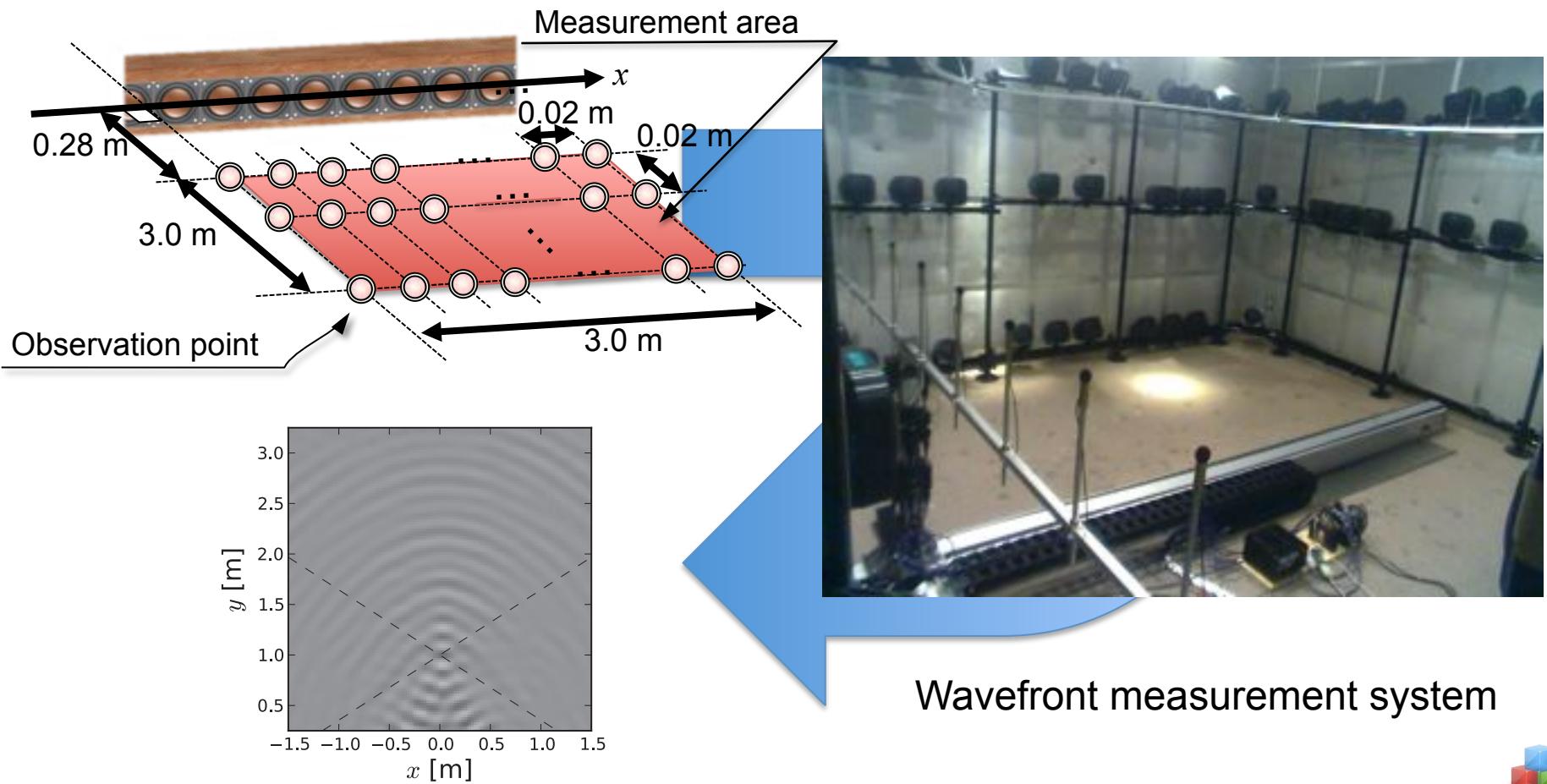
Fig. Constructed loudspeaker array

$$f_{alias} = \frac{c}{2\Delta x} \simeq 2.5 \text{ kHz} \quad (c : 343.26 \text{ m/s in } 20^\circ\text{C})$$

This value higher than the upper frequency band limits of **measure cues of sound localization**

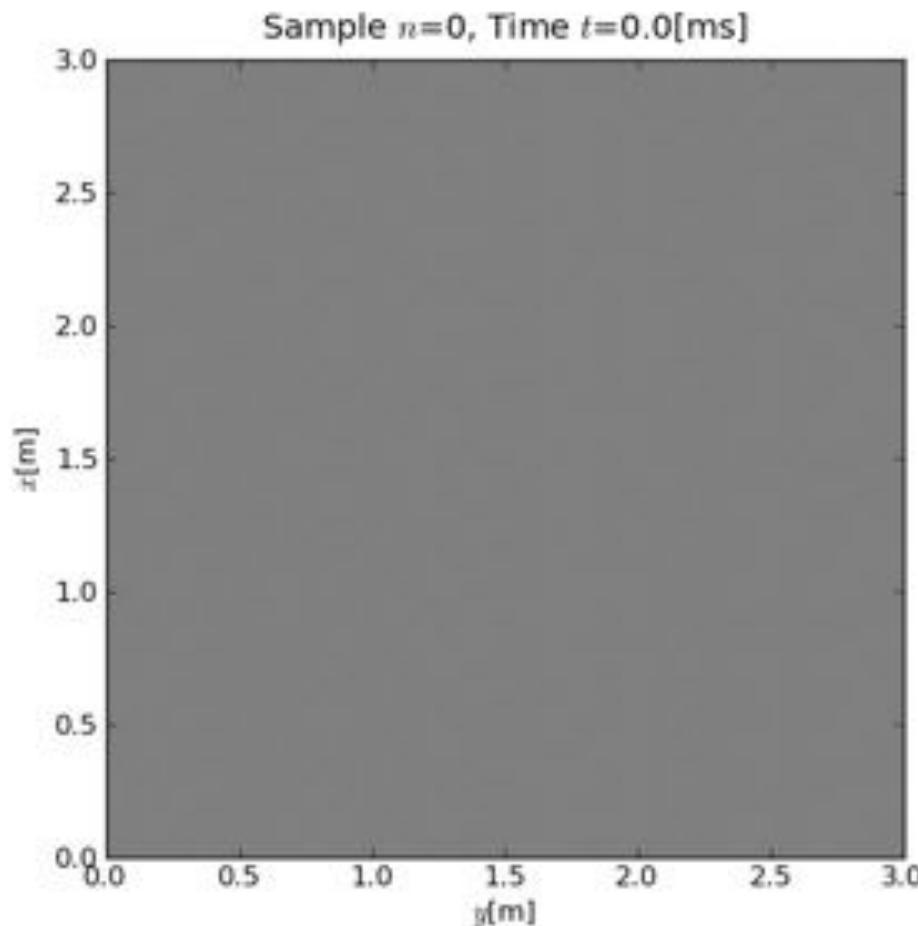
# ライブラリを使用したアプリの一例

- Observation area become restricted by array microphone interval and number

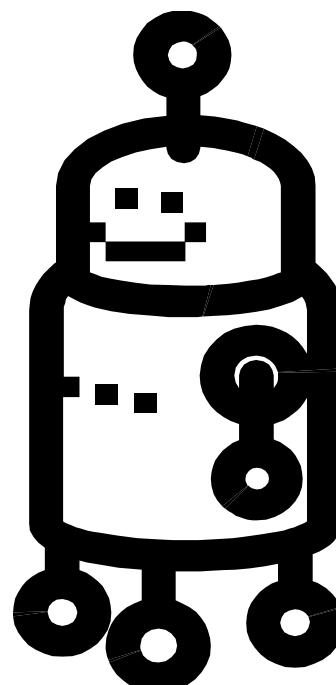


# ライブラリを使用したアプリの一例

空間を伝播する音を可視化することができる



そんな便利なライブラリを使って  
Kinect を使うとこうなる



# Kinect を使用したマルチモーダル処理の一例

## ■ 方位別学習済み ICA フィルタデータベース

- ⇒ 11.4°ごとにDBを用意
- ⇒ Kinectの画角を5等分

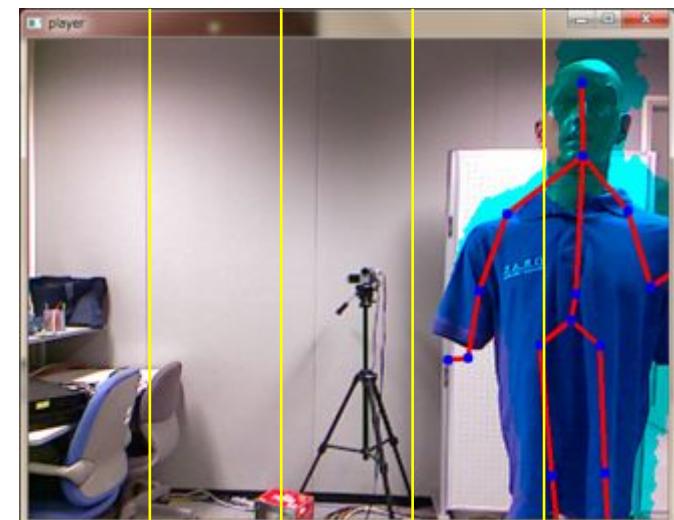
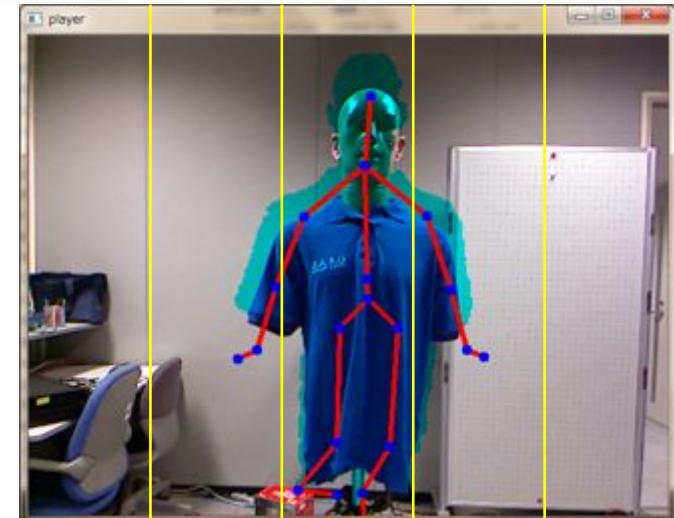
## ■ 実環境でリアルタイムに人形を認識

- ⇒ 20 フレーム/s
- ⇒ 頭部位置情報をトラッキング
- ⇒ 学習済み ICA フィルタ DB よりロード

BSSA初期化

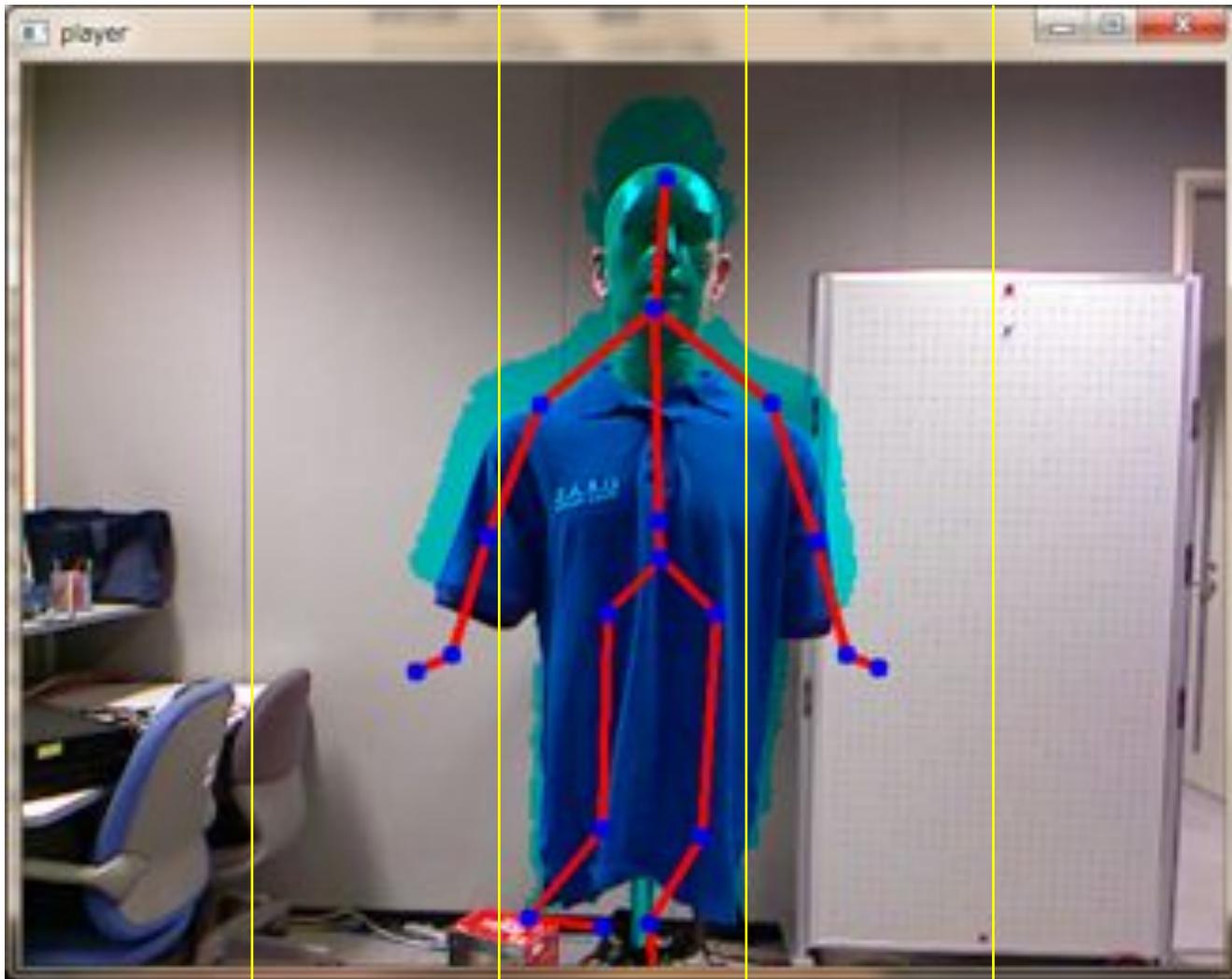


JNAS1文認識



# 何よりも驚いたこと

- ケマーが人として認識される！（実験の再現性を保証可能）



# Kinect によるリアルタイム雑音抑圧

## ■ PCスペックのお陰でリアルタイムに動作

⇒ (アルゴリズム遅延 50 ms)

## ■ こんな安いデバイスで本当にうまくいくの？

### 動作環境

CPU	Intel core i7 1.87 [GHz] (4core, 8thread)	使用メモリー	35 [MB]
メモリー	DDR3 8 [GB]	CPU 平均使用率	20% (2core分に相当)
OS	Windows7 64 bit		

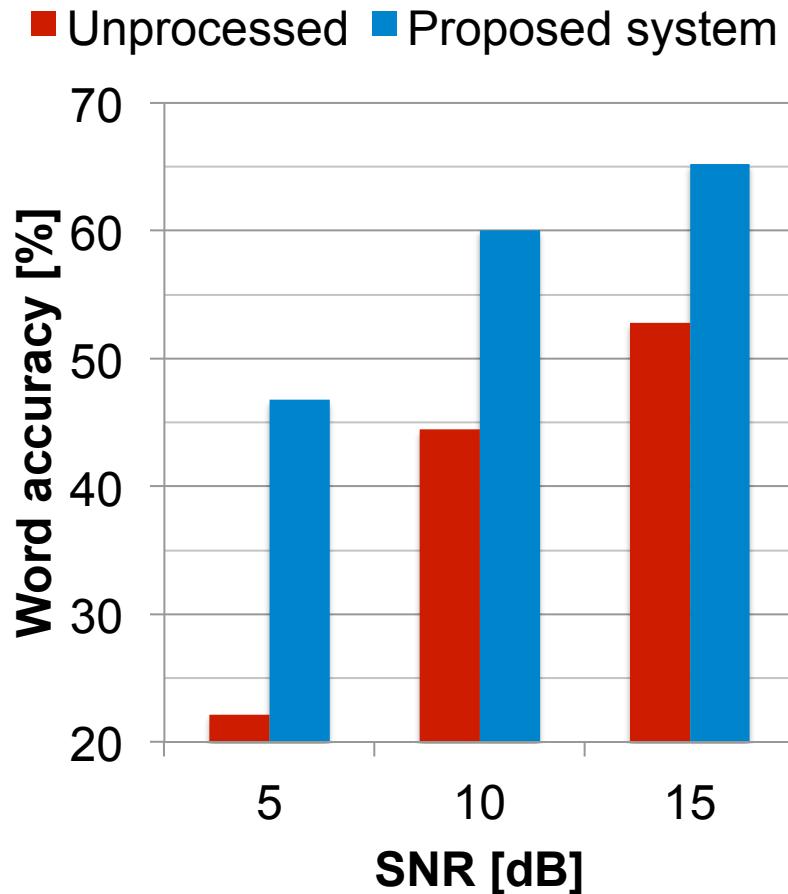
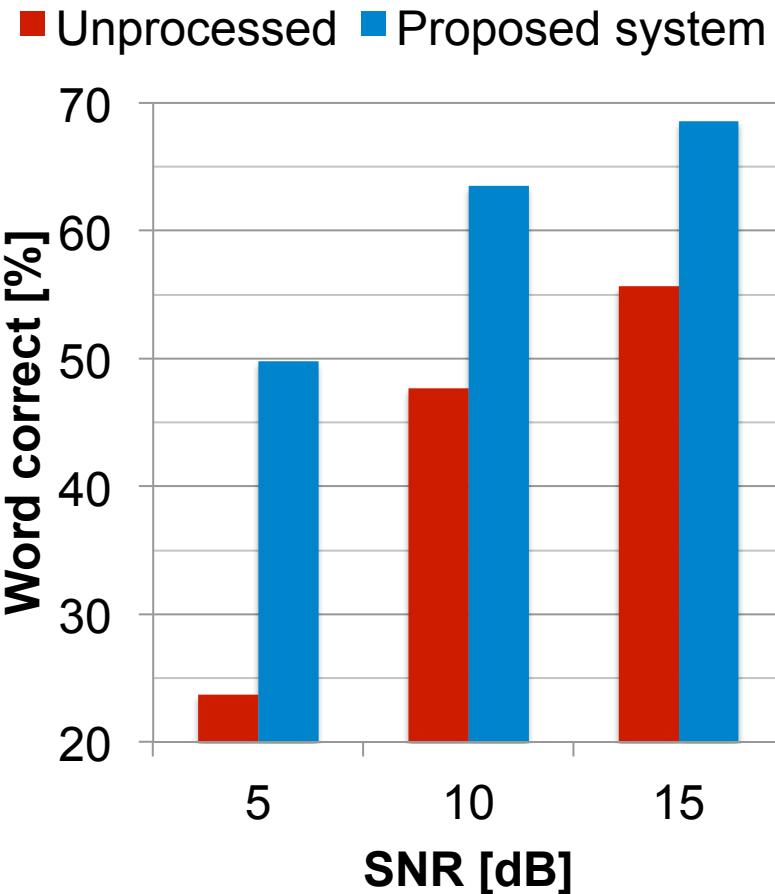
リアルタイムデモ音

観測信号 

目的音推定 



# 音声認識実験結果



- Kinect のような安いデバイスでも音声認識実験に使える
- すべての条件において、提案法により10%以上の改善が確認できる

# デモ

## 駅構内案内システム「キタちゃん」との接続デモ



[http://spalab.naist.jp/kinect\\_bss\\_demo.html](http://spalab.naist.jp/kinect_bss_demo.html)

# まとめ

- 安いデバイス (Kinect) と便利なライブラリで夢のある話をした
- プログラムが苦手な私でもデモを 1ヶ月程度で作れた
- ヒント
  - ⇒ port audio
  - ⇒ libsndfile
  - ⇒ python (numpy, scipy, matplotlib, pyside)
  - ⇒ ASIO4ALL (Windows ユーザー)
- 詳細は若手フォーラムで今後公開予定！
- 理論も実装もできる楽しい研究ライフを若手フォーラムで！



■ これだけだと夢のままでですが

若手フォーラムに入ると  
夢が現実に！