Vol.2016-MUS-111 No.22
2016/5/21

ロボット聴覚クラウドサービス HARK SaaS の紹介と音環境分析アプリケーションの開発

水本 武志[1,a)]

概要：マイクロホンアレイで収録された音響信号から音源定位や音源分離を行う手法が実装されたロボット聴覚ソフトウェア HARK (Honda Research Institute Japan Audition for Robots with Kyoto University) が 2008 年より公開されている。マイクロホンアレイ技術の利用は従来より容易になったものの、その利用には、計算機環境の要求スペックの高さやインストールの手間などのハードルが依然あった。本稿では、この問題を解決する、インターネット経由で利用できるクラウドサービス HARK SaaS (Software as a Service) とそのアプリケーションを紹介する。HARK SaaS は、ブラウザで利用できるインタフェースや Web API を備えているので、音環境の分析や既存ソフトウェアへの組み込みが従来よりも容易となる。

1. はじめに

ロボット聴覚ソフトウエア HARK とは、2008 年から公開されている音源定位・音源分離などのマイクロホンアレイ技術が実装されたソフトウェアである [1]。公開以来 HARK は様々なシステム、例えばクイズの司会 [2] やテレプレゼンスロボット [3] に応用されてきた。しかし、依然として、インストールの手間が大きい、計算機のスペック要求が高いなどの利用上のハードルがあった。

そこで我々は、HARK をネットワーク経由で利用できるように実装したクラウドサービス HARK SaaS を開発したので報告する [*1]。本サービスの基本的な機能は、ユーザがマイクロホンアレイで録音した多チャンネル音響ファイルをアップロードすると、HARK によるマイクアレイ処理を行い、その結果を JSON 形式 で返すことである (図 1)。本サービスは Web インタフェースと Web API を提供しているので、ブラウザ上の操作や HTTP リクエストによって、従来より容易にマイクロホンアレイ処理技術を利用できる [*2]。たとえば、ネットワーク接続できる小型計算機 (Raspberry Pi® (Raspberry Pi Foundation) や BeagleBoard ™ (テキサス・インスツルメンツ社、Digi-Key) など) にマイクを接続すれば、すぐに利用を開始できる。なお、小型計算機上で HARK を実行する試みもあり、専用の実装による車載 HMI [5] や、両耳聴処理によるロボット制御 [6] の報告がある。

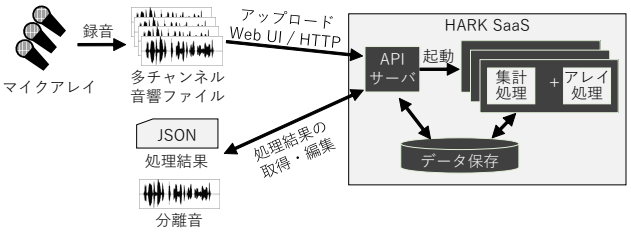

図 1 HARK SaaS の概要

2. 音のクラウドサービス

音声や音楽などの音データを利用するクラウドサービス (以下、音のクラウドサービスと呼ぶ) は複数公開されているが、本サービスのように多チャネル音響データから音環境情報を抽出するものは我々の知る限り存在しない。たとえば、SoundCloud® [*3] や YouTube® [*4] は主にアップロードされた音データ自体を他のユーザと共有するためのサービスである。また、Apple 社の音声対話サービス Siri®、音声認識と音声合成サービス Rospeex [7]、Google Speech API などは、人の音声の認識や合成、すなわちテキスト情報、に特化したサービスである。一方、能動的音楽鑑賞サービス Songle[8] は、アップロードされた音楽音響信号からサビ区間やメロディなどの情報を抽出する点で本サービスに類似している。

[1] (株) ホンダ・リサーチ・インスティチュート・ジャパン
[a)] t.mizumoto@jp.honda-ri.com
[*1] 本稿は [4] の要約版である。
[*2] https://api.hark.jp で期間限定で実験的に公開中。
[*3] https://soundcloud.com
[*4] https://www.youtube.com

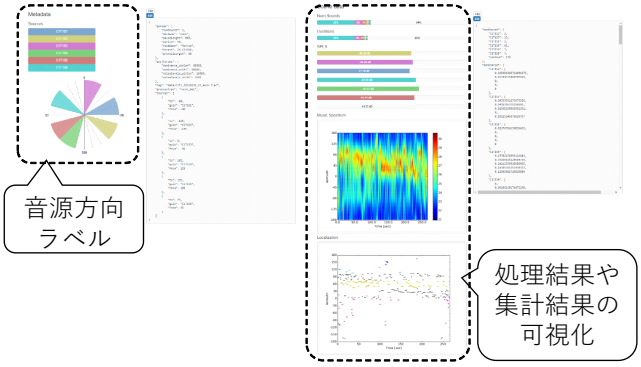

図 2　Web インタフェースの可視化機能

```
import pyhark.saas
h = pyhark.saas.PyHarkSaaS("API_KEY", "API_SECRET")
h.login()                                # 認証
h.createSession(metadata)                # パラメータ設定
h.uploadFile(open(filename, 'rb'))       # ファイル送信
h.wait()                                 # 処理終了待ち
result = h.getResults()                  # 処理結果受信
```

図 3　HARK SaaS SDK を用いたサンプルコード

3. HARK SaaS の設計と実装

3.1 サービスの設計

HARK SaaS 設計上の課題は次の 3 点である。

インタフェースの汎用性
他のシステムとの統合を容易にするため、標準規格に準拠した汎用的なインタフェースが必要である。

ユーザビリティ
プログラムを作成せずに機能を利用したいユーザとシステムを開発したいユーザの両方に利用しやすいユーザインタフェースが必要である。

信頼性
安心して利用できるようにするため、セキュリティや安定性の向上が必要である。

まず、インタフェースの汎用性については、本サービスの全機能について HTTPS リクエストで操作できる Web API を提供し、データの送受信は JSON フォーマットで行う。ほぼあらゆるプログラミング言語が HTTPS リクエストと JSON フォーマットをサポートしているので、本設計によってインタフェースの汎用性が確保できたと言える。

次に、ユーザビリティについては、各ユーザ層ごとにインタフェースを提供する。まず、プログラムを作成せずに利用できるようにするため、HARK の一連の処理をブラウザから実行できる Web インタフェースと、図 2 に示す実行結果可視化機能を提供する。次に、システムを開発したいユーザに対しては、上記の Web API に加えて、Python で実装した HARK SaaS SDK (Software Development Kit) を提供することで簡単に統合できる手段を提供する。本 SDK のサンプルコードを図 3 に示す。

最後に、信頼性は、2 つの観点からの対策が必要となる。第 1 にセキュリティについては、次の対策を実施した。

(1) HTTPS プロトコルを用いて通信を暗号化する。
(2) ユーザ毎に発行する認証キーを発行し、サービスの利用には毎回変化する一時認証トークンを要求する。
(3) データ保存サーバと外部からアクセスできるサーバを分離し、サーバに侵入時のデータ漏洩を防ぐ。
(4) ユーザ管理は Google OAuth を用い、サービス内にパスワードを保存しない。

第 2 に、安定性については、サーバを役割ごとに分類して複数台用意し、負荷分散によって大規模アクセスがあっても安定的に動作する設計を行った。

3.2 データ構造の設計

本サービスでは、ひとつの音響ファイルを処理単位と定義し、**セッション** と呼ぶ。すべての処理結果やパラメータはセッション単位で表現する。具体的には、ひとつのセッションに対して、次の 3 種類のデータを提供する。

メタデータ
ユーザが与えるデータ。例えば、信号処理パラメータや音源方向ラベルが含まれる。音源方向ラベルとは、3 つの情報 (範囲の開始角度, 終了角度, ラベル名) の配列で、音源定位された方向がいずれかの音源方向ラベルの範囲内に含まれる場合、これに対応するラベル名が付与される。たとえば、マイクロホンアレイと音源の位置関係が変化しない場合 (会議など) に、音源定位された音イベントにラベルを自動付与できる。

コンテキスト情報
音イベント毎のデータ。MUSIC 法による音源定位の出力を単位とし、定位された音イベント毎に定義する。例えば、音イベントの開始時間と終了時間、仰角と方位角、音量、分離音などが含まれる。

シーン情報
セッション全体に対して定義するデータ。音響ファイル自体の情報やコンテキスト情報を集計した結果などを含む。例えば、処理される音ファイルの長さやサンプリングレート、音量時系列等の情報や、音源方向ラベルごとの音イベント数、その合計時間などの音源方向ラベルごとの情報が含まれる。

3.3 サービス実装

本サービスは Amazon Web Services (AWS) 上に実装した。データベースや負荷分散機構等、サービスの複数の部分に AWS のサービスを用いることで、運用者の負荷の軽減と安定性確保の両立を実現した。

また、並列で関数を実行できる AWS Lambda の活用によって、処理速度も向上した。これは、HARK SaaS では

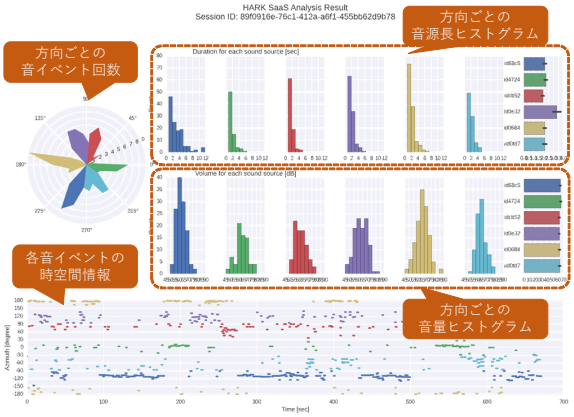

図 4　HARK SaaS サンプルアプリケーション： 音環境可視化

HARK による音響処理だけでなく、分離音のファイルサーバへの送信やデータベースへの登録などの後処理が必要となることが原因である。並列化によって、後処理に要する時間を削減できたことで、全体の処理時間が短縮できた。たとえば、30 分の 8 チャネル音響データを処理する場合、並列実行なしでは処理時間は 30 分程度必要であったが、並列実行を行うと その 30% 以下の時間 (約 8 分-9 分) で処理し、結果を受け取ることができる。

3.4　HARK SaaS サンプル：音環境可視化

HARK SaaS の応用例として、音環境可視化アプリケーションを紹介する (図 4)。本アプリケーションは、録音された音ファイルを HARK SaaS へアップロードし、処理結果を受信し、処理結果とメタデータで設定された音源方向ラベルを利用して結果を可視化する。本アプリケーションは音源方向ラベルごとの音イベント集計結果を色分けして表示するので、方向ごとの音環境分析ができる。なお、サンプルには HARK SaaS の Python SDK と、可視化ライブラリ Seaborn、 Matplotlib を使用した。

可視化画面は 4 部分から構成されている。まず、図左上は方向ごとの音イベント数を表し、音源方向ラベルごとに色分けがされている。この図から、音源方向ラベルの方向に関する傾向が分かる。例えば図 4 の場合、緑色の音源方向ラベルの音イベントは 0 度 方向から多く発生していることがわかる。次に、図右上部は音源方向ラベルごとの音イベントの継続時間のヒストグラムと平均値を表す。ヒストグラムから音源方向ラベルごとの継続時間の傾向を分析でき、右端の平均値から音源方向ラベル同士の継続時間の比較ができる。続いて、図右中部は音源方向ラベルごとの音イベント音量のヒストグラムと平均値を表す。ここでも継続時間と同様に音源方向ラベルごとの分析やそれぞれの比較ができる。最後に、図下部は時間・方向ごとの音イベントを表す。この図より、-120 度の方向からは 100 秒から 180 秒、280 から 380 秒、 430 秒から 500 秒の 3 回にわたって音イベントが連続的に発生していることがわかる (濃青で表示)。

4.　まとめ

本稿では、ロボット聴覚ソフトウェア HARK のクラウドサービス版である HARK SaaS の紹介を行った。本サービスを使うことで、従来のローカル型 HARK より簡単にマイクロホンアレイ処理を使用したり、それを組み込んだシステムの実装が容易になると期待できる。今後は、音源同定をはじめとした音源定位・分離以外の機能の拡張などを行う予定である。

謝辞　サービスの実装に協力して頂いた菅原哲也氏に感謝する。

参考文献

[1] Nakadai, K., Okuno, H. G., Nakajima, H., Hasegawa, Y. and Tsujino, H.: Design and Implementation of Robot Audition System “HARK”, *Advanced Robotics*, Vol. 24, pp. 739–761 (2009). doi:10.1163/016918610X493561.

[2] Nishimuta, I., Yoshii, K., Itoyama, K. and Okuno, H. G.: Toward a Quizmaster Robot for Speech-based Multiparty Interaction, *Advanced Robotics*, Vol. 29, No. 18, pp. 1205–1219 (2015).

[3] Mizumoto, T., Nakadai, K., Yoshida, T., R. Takeda, T. Otsuka, T. T. and Okuno, H. G.: Design and Implementation of Selectable Sound Separation on the Texai Telepresence System using HARK, *ICRA*, pp. 694–699 (2012).

[4] 水本武志，中臺一博：HARK SaaS: ロボット聴覚ソフトウェア HARK のクラウドサービスの設計と開発，人工知能学会 AI チャレンジ 研究会 (第 43 回)，pp. 60–65 (2015).

[5] 中臺一博，水本武志，中村圭佑：モバイル端末用マクロホンアレイシステムの開発とコミュニケーション支援への適用，ロボット学会学術講演会 (2015).

[6] 坂東宜昭，金　宜鉉，糸山克寿，吉井和佳，中臺一博，奥乃　博：両耳聴ロボット聴覚ソフトウェア HARK-Binaural の紹介と Raspberry Pi 2 を用いたヒューマノイドロボットへの適用，音学シンポジウム (2015).

[7] 杉浦孔明，堀　智織，是津耕司：rospeex:クラウド型音声コミュニケーションを実現する ROS 向けツールキット，電子情報通信学会技術研究報告，クラウドネットワークロボット，Vol. 113, No. 248, pp. 7–10 (2013).

[8] Goto, M., Yoshii, K., Fujihara, H., Mauch, M. and Nakano, T.: Songle: A Web Service for Active Music Listening Improved by User Contributions, *ISMIR*, pp. 311–316 (2011).