音声対話向けクラウドロボティクス基盤 rospeex の構築と長期実証実験

○杉浦孔明，堀智織，是津耕司（情報通信研究機構）

1. はじめに

スマートフォンを始めとする種々のデバイスに音声インタフェースが導入され，広く一般に認知されるようになってきた [1, 2]．音声対話システムの分野では，開発者が容易に利用できるツールキットも公開されている（例えば [3]）．一方，人とロボットのインタラクションでは，高性能な音声認識・合成を容易に利用できる状況ではない．ロボットとの高度な音声インタラクションを可能とするためには，音声処理とロボティクスの深い知識を要求されるのが現状である．

本研究では，家庭やオフィス内で「テーブルの上のペットボトルを取って」などの音声対話を行うサービスロボット開発を想定する．代表的なタスクとしては，ロボカップ@ホーム [4] タスクが挙げられる．このようなタスクでは，開発コストを低減できることから，RT ミドルウェアや ROS（Robot Operating System）などのミドルウェアの利用が一般的になってきている．ミドルウェアに対応するソフトウェアとして，音声認識・対話・合成を可能にするモジュールが提供されているものの，スタンドアロン型を前提とするものが多い [5,6]．しかしながら，ロボットに搭載できるコンピュータにはストレージやメモリの制限があるため，高性能な音声認識・合成は難しい．一方，クラウド型の高性能な音声認識・合成 API を提供する企業もあるが，ロボット開発者を想定したものでない場合が多い．

そこで本研究では，クラウドロボティクス基盤 “rospeex” を構築・公開し，サービスを長期間にわたり実運用した．音声認識および音声合成機能をクラウド化することで，音響モデルや言語モデルなどの大規模な資源をロボット上に搭載する必要がなくなり，ハードウェアを簡略化することでコストを低減できる．クラウド型音声認識・合成サービスを通じて，NICT で開発されたエンジン[1] をユーザは利用可能である．また，他のクラウド型音声認識・合成サービスに切り替えて利用することも可能である．

クラウドロボティクスやクラウドネットワークロボティクスなどの分野では，物体認識，知識共有，機械学習などのためにクラウドコンピューティングを用いるアプローチが提案されている [7–10]．本研究はこれらと関連するが，ロボットの音声コミュニケーションに主眼を置く点が異なる．また，HARK [6] や OpenHRI [5] などミドルウェアに対応した音声コミュニケーションツールでは，内部的に Julius [11]，Festival [12]，OpenJTalk [13] などスタンドアロン型のエンジンを用いている．これらのエンジンは機能的には複数言語の音声認識・合成が可能であるが，言語モデルの入れ替えなどをロボット開発者自身が行う必要がある．一方，提案手法では，次節で説明するように言語やボイスフォントの変更を簡単に行うことができる．図 2 に rospeex による音声対話の例を示す．

本研究の独自性は以下である．

- 音声対話タスクを目的としたクラウドロボティクス基盤を構築・公開し，長期実証実験を行った．

2. rospeex の機能

rospeex が提供する機能と想定する標準的な構成を図 1 に示す．ROS のディストリビューションとしては，Groovy Galapagos を想定する．発話理解（言語理解），対話制御，応答生成については，ユーザが記述するものとする．

2·1 雑音抑圧および発話区間検出

ロボットとの音声対話において特徴的な難しさは，push-to-talk 方式を前提とできないことである．スマートフォン型音声インタフェースなどでは，音声の入力前にボタン等のインタフェース（「OK，XXX」のように音声の場合もある）を用いることが多い．これに対し，例外はあるものの，ロボットではそのようなインタフェースを前提とすることはできない．したがって，発話区間検出（Voice Activity Detection; VAD）が重要になる．さらに，マイクと話者の距離が大きいことが多いため，雑音抑圧が必要である．ロボカップ@ホームのような高騒音環境では，雑音抑圧を使用しなければ正確な発話区間検出はほぼ不可能である．これがヘッドセットを前提としたロボット以外のシステムとの大きな違いである．以下，本論文では，rospeex の使用者を「ユーザ」，ロボットとの対話者を「話者」と略す．

rospeex では，雑音抑圧と発話区間検出はネットワー

図 2rospeex を用いたロボット対話の例

[1]rospeex 以外では，音声翻訳サービスとして既に 1000 万発話を処理している [5].

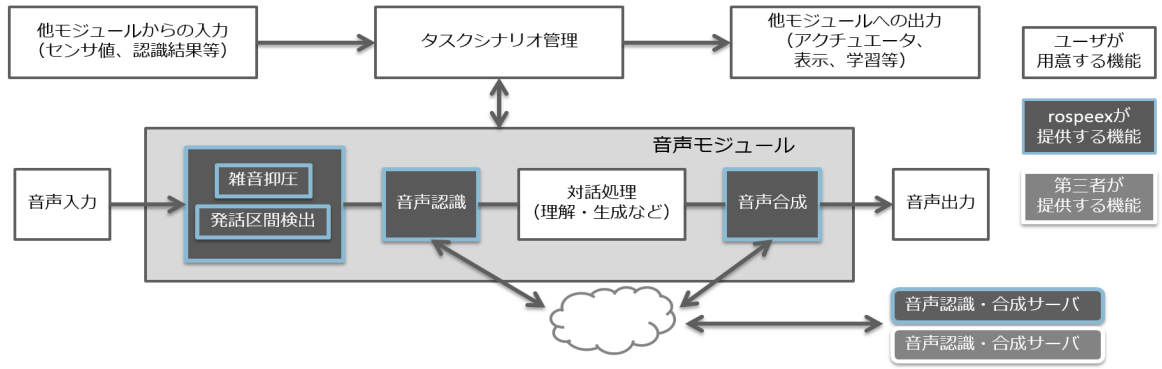

図 1rospeex の構成の概略

ク上サーバで行わない設計としている．これらをサーバで処理するとネットワーク由来の遅延によりリアルタイム性の確保が難しくなるためである．また，一般的に発話区間検出の精度はそれほど高くないため，後段の処理でロボット名を含む発話のみ受け付けるなどの工夫が必要である．

2·2 クラウド型音声認識・合成

rospeex は複数のクラウド型音声サービスに接続可能であり，それらを切り替えて使用できる．本節では，NICT が提供する音声認識・合成サービスについて説明する．これらは，ROS を経由せずに単体としても利用可能であり，4 か国語（日英中韓）の音声認識・合成に対応している．現時点では，学術研究目的に限り無償・登録不要で公開している．本サービスでは，JSON ファイルをインタフェースとする．ユーザが用いるプログラミング言語には依存しないため，C++や Python など各種のプログラミング言語を利用可能である[2]．

日本語の音声合成については，非モノローグ HMM 音声合成 [14] に対応している．一般的な音声合成器は人-ロボット対話に最適化されている訳ではないが，非モノローグ音声合成を選択することでロボットとの対話に特化して開発されたボイスフォントを利用可能である．非モノローグ音声合成手法の性能評価については，[14] を参照されたい．

2·3 ロボット開発者向けの API

rospeex の対象とするユーザはロボット開発者であるため，音声認識・合成を簡単に使用できることが望ましい．前述したように，NICT のサーバではインタフェースに JSON を用いている．rospeex では，開発者が 1 行程度の記述で音声認識・合成を利用可能な API を用意している．Python や C++であれば，10 行程度で簡単な対話（時刻の問い合わせなど）を行う関数を記述することができる．

対話管理にマークアップ言語（VoiceXML など）を利用するソフトウェアと異なり，rospeex は対話管理の簡単なインタフェースを用意していない．これは，想定ユーザとして，複雑な対話管理を必要としないロボット開発者を念頭に置いたためである．一方，ROS 上で Python や C++で開発したソフトウェア資産があれば，rospeex と簡単に組み合わせることが可能であるという利点がある．また，現状では，音源定位などの音響処理は統合されていない．しかしながら，HARK [6] など音響処理を扱うモジュールが提供されているので，rospeex の前段に容易に組み込むことが可能であると考えられる．

[2]サンプルコードを http://komeisugiura.jp/software/nm_tts.html から入手可能である．

3. 実証実験

3·1 実験設定

本節では，処理速度の面から提案システムの実用性を評価する．これは，クラウド型の処理速度が実用的な範囲でなければ，スタンドアロン型を用いることが合理的であるためである．2013/12/1 から 2014/5/31 までのアクセス記録をもとに，実際の利用における処理時間を分析した．評価を音声認識と音声合成に分けて行なう．

音声合成の品質評価については，[14] を参照されたい．実験に用いたサーバの CPU は Intel 製 X5690（12 コア，3.47GHz），メモリは 200GB であった．

3·2 結果

表 1 にリクエスト数，処理時間，リアルタイムファクター（RTF）を示す．ここに，RTF は以下で定義される．

$$\mathrm{RTF} = \frac{T_p}{T_u} \tag{1}$$

ここに，T_p と T_u は，それぞれ処理時間と音声の長さを表す．

表 1 より音声認識の RTF は 0.76 程度である．これは，音声認識処理に発話の 76%程度の時間を要したことを

表 1 2013/12/1〜2014/5/31 の期間におけるサーバの処理時間．

タスク	音声認識	音声合成
リクエスト数	10445	23519
処理時間（中央値）[msec]	961	399
RTF（中央値）	0.760	0.192

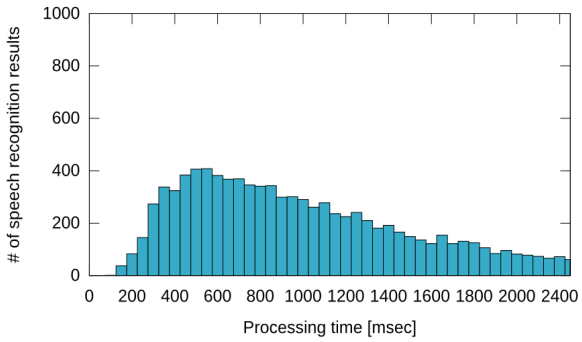

図 3 処理時間のヒストグラム（音声認識）

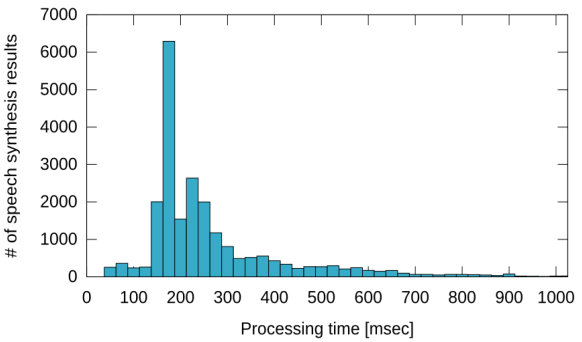

図 4 処理時間のヒストグラム（音声合成）

示している．サーバサイドの処理時間としては十分な速度であるが，実際にはクライアントサイドで発話検出完了後にサーバに音声を送出していることから，総合的な処理時間としては十分に速いとはいえない．今後の課題としては音声の分割送信による（総合的な）高速化が挙げられる．理論的には発話検出が完了したと同時に音声認識結果を得ることができる．一方，音声合成の RTF は 0.2 程度であるので，合成された音声の 1/5 程度の長さの時間で合成処理が可能であった．

図 3 に，実証実験においてサーバが音声認識に要した時間の分布を示す．処理時間の中央値は 961[msec] であるが，2000[msec] 程度まで幅広く分布している．これは，入力された発話の長さにほぼ比例している．1000[msec] 以上の処理時間はスムーズな対話を妨げる可能性があるため，音声の分割送信などにより高速化を進める必要がある．

図 4 に，音声合成に要した時間の分布を示す．図より，多くの発話は 400[msec] 以内に処理が終了していることがわかる．ただし，数は少ないものの，処理が 1000[msec] 以上かかる場合も存在する．長文のテキスト入力（ニュースサイトの本文のコピーなど）が行われる場合があったためである．これは，サンプルとしてブラウザからのテキスト入力インタフェースを用意したことにより，容易に長文を入力できたためであると考えられる．ただし，入力は複数の文から構成されていたため，適切に分割すれば全体として待ち時間を大幅に短縮できると考えられる．以上のように，本実験ではネットワーク遅延が考慮されていないものの，実用的に十分な処理時間であったといえる．

4. おわりに

ネットワーク接続を前提とすれば，音声認識･合成処理をクラウド化することで，ロボットの低コスト化が可能である．本研究では，音声対話向けのクラウドロボティクス基盤 rospeex の構築と長期実証実験を行なった．本基盤の応用としては，サービスロボット･音声対話システムの開発，音声の書き起こし，文書の読み上げ，などが挙げられる．なお，rospeex は `http://rospeex.org` からダウンロード可能である．

謝辞

本研究の一部は，科研費（若手 (B)24700188）の助成を受けて実施されたものである．

参考文献

[1] 河原達也，“音声対話システムの進化と淘汰–歴史と最近の技術動向–,” 人工知能学会誌，vol.28，no.1，pp.45–51，2013.
[2] 松田繁樹，林輝昭，葦苅豊，志賀芳則，柏岡秀紀，安田圭志，大熊英男，内山将夫，隅田英一郎，河井恒，中村哲，“多言語音声翻訳システム“ VoiceTra ”の構築と実運用による大規模実証実験,” 電子情報通信学会論文誌，vol.J96-D，no.10，pp.2549–2561，2013.
[3] 大浦圭一郎，山本大介，内匠逸，李晃伸，徳田恵一，“キャンパスの公共空間におけるユーザ参加型双方向音声案内デジタルサイネージシステム,” 人工知能学会誌，vol.28，no.1，pp.60–67，2013.
[4] 杉浦孔明，“ロボカップ@ホームリーグ,” 情報処理，vol.53，no.3，pp.250–261，2012.
[5] 松阪要佐，“OpenHRI,” 日本ロボット学会誌, vol.31, no.3, pp.2–7, 2013.
[6] K. Nakadai, T. Takahashi, H.G. Okuno, H. Nakajima, Y. Hasegawa, and H. Tsujino, “Design and Implementation of Robot Audition System’HARK’—Open Source Software for Listening to Three Simultaneous Speakers,” Advanced Robotics, vol.24, no.5-6, pp.739–761, 2010.
[7] R. Arumugam, V.R. Enti, L. Bingbing, W. Xiaojun, K. Baskaran, F.F. Kong, A.S. Kumar, K.D. Meng, and G.W. Kit, “DAvinCi: A Cloud Computing Framework for Service Robots,” Proc. ICRA, pp.3084–3089, 2010.
[8] B. Kehoe, A. Matsukawa, S. Candido, J. Kuffner, and K. Goldberg, “Cloud-Based Robot Grasping with the Google Object Recognition Engine,” Proc. ICRA, 2013.
[9] K. Kamei, S. Nishio, N. Hagita, and M. Sato, “Cloud Networked Robotics,” Network, IEEE, vol.26, no.3, pp.28–34, 2012.
[10] M. Tenorth, A.C. Perzylo, R. Lafrenz, and M. Beetz, “The RoboEarth Language: Representing and Exchanging Knowledge about Actions, Objects, and Environments,” Proc. ICRA, pp.1284–1289, 2012.
[11] http://julius.sourceforge.jp/
[12] http://www.cstr.ed.ac.uk/projects/festival/
[13] http://open-jtalk.sp.nitech.ac.jp/
[14] 杉浦孔明，志賀芳則，河井 恒，翠 輝久，堀 智織，“サービスロボットのための非モノローグ hmm による音声合成,” 第 31 回ロボット学会学術講演会資料，pp.2C1–02，2013.