多言語の TTS に対応した Lip Sync ライブラリの作成

4K−4

村上 秀幸† 馬場 博巳‡ 乃万 司†
†九州工業大学情報工学部 ‡近畿大学九州工学部
{h_mura,noma}@pluto.ai.kyutech.ac.jp baba@fuk.kindai.ac.jp

1 はじめに

発話時における口の動きは，発話内容を理解する上で非常に重要である．そのため人体アニメーションにおいても，発話にあわせてエージェントの口を動かすことが望ましい．これを Lip Sync と呼ぶ．エージェントに任意の発話をさせる必要があるとき，音声生成には一般に Text-To-Speech (TTS) システムが用いられるが，その過程で生成される音素情報を用いた Lip Sync は従来から実現されている．しかし従来，その実装には TTS 固有の API を用いる必要があり，また，口の動きの計算法も単一ではないため，多種類の TTS と口の動きの計算法とを組み合わせて用いるのは困難であった．

そこで我々は，TTS と口の動きの計算法との独立性を高め，任意の組み合わせによる Lip Sync を可能とするライブラリ (LSSLib) を開発した．なお，今回は MS Windows98 上で，Visual C++ を用いて開発した．

2 ライブラリの構成

複数種類，複数言語の TTS と口の動きをサポートするライブラリを実現するには，(1) TTS と口の動きの計算部分を外部から隠蔽し，(2) それらの変更や追加を容易に行えるようにし，TTS ごとの機能の違いを吸収したインタフェースを提供する必要がある．そこで本研究では，LSSAgent，TTSManager，CALManager の 3 つのモジュールから構成されるライブラリを実現した．

A Lip Sync Library for Multi TTS Environments
Hideyuki MURAKAMI†, Hiromi BABA‡, and Tsukasa NOMA†
†Kyushu Institute of Technology
‡Kinki University

2.1 LSSAgent モジュール

LSSAgent モジュールは Lip Sync を行うシステム開発に必要なインタフェースを提供する．また，1 人のエージェントが使用する言語ごとの TTS の情報と口の動きの計算法の情報を管理する．さらに，前後の音素情報を参照して口の動きを計算する場合 [1] も想定し，あらかじめ TTS を用いテキストを音素情報へ変換させた情報を蓄え管理する機能を持つ．

2.2 TTSManager モジュール

TTSManager モジュールはシステム内で使用するすべての TTS を管理する．また，多種類 (多言語) の TTS は異なる API を持ち，さらにその機能も一般に異なるため，TTS のインタフェースを統一し，それらの機能の違いを吸収する．これにより TTS の機能の違いを考慮せずに，共通のインタフェースで TTS の操作が可能となる．さらに TTS の本体を動的リンクライブラリ (DLL) の形式で外部から提供することで，TTS の変更や追加を容易に行えるようにしている．

音素集合の違いの吸収． TTS は各々異なる音素集合を使用しているため，生成される音素情報も異なる．そのため口の動きを計算するために音素情報を必要とするとき，それがどの TTS から生成されたものかを知る必要があり，それらに応じた実装をしなければならならない．すべての TTS で使用される音素集合を標準化することも考えられるが，各言語ごとに特徴的な発音が存在するため困難である．そこで今回は，各言語ごとに音素集合を標準化し，TTS から生成される音素情報をこの標準音素集合を用いたものに変換するようにした．これにより，口の動きの計算法の実装を，TTS 単位から言

語単位へと単純化することができた.

2.3 CALManager モジュール

CALManager モジュールは，システム内で使用するすべての口の動きの計算法を管理する．また，インタフェースを統一することで，複数の計算法を外部からは同一に扱うことを可能としている．さらにその計算部分の本体は DLL の形式で外部から提供することで，変更や追加を容易に行えるようにしている．

3 本ライブラリの使用例

本ライブラリのプログラム例を図 1に示す．

エージェントの初期化. まず，(a) エージェントを作成する．次に (b) 使用する TTS を識別子と音声の種類をもとに作成し，それをエージェントに割り当てる．そして，(c) 使用する口の動きの計算法を識別子と初期化ファイル名をもとに作成し，同様にエージェントに割り当てる．

発話要求時. エージェントに発話をさせる場合には，(d) まず言語を指定し，その言語に割り当てられている TTS を使用して，一旦テキストを音素情報へと変換する．次にその音素情報を使用して音声を生成し出力する．

画像生成時. 画像を生成する場合は，(e) 現在割り当てられている口の動きの計算法を使用し，現在の口の形状パラメータを取得する．次にそのパラメータをもとにエージェントの画像を生成する．

4 本ライブラリの特長

本ライブラリには，前述したようにプログラムを簡潔に実装できることの他に，実行時に TTS や口の動きの計算法を容易に切替えることができるという特長がある．

まず，TTS の切替えが容易なことから，エージェントが単一の言語を発話するばかりではなく，例えば，英語と日本語を自由に切替えて発話できるようになる．

一方，口の動きをリアルにするほどその計算コストは大きくなる．また，仮想人間エージェントに要求される口の動きのリアルさは，エージェントとカメラの距離に依存する．これらのことから，ウォークスルーシステムなどでは画面上のエージェントの大きさに応じて口の動きの計算法を切替えることが望ましい．これは Lip Sync における一種の LOD(Levels Of Detail) であり，本ライブラリを用いることにより簡単に実現できる．

```
LSSAgent agent;                                    (a)
        ⋮
TTSINFO ttsi;
strcpy(ttsi.EngineName,"TTSの識別子");
strcpy(ttsi.VoiceName,                             (b)
        "使用する音声の種類");
agent.CreateTTS(JAPANESE,ttsi);
        ⋮
CALINFO cali;
strcpy(cali.EngineName,
        "口の動きの計算法の識別子");
strcpy(cali.InitFile,"初期化ファイル名");          (c)
agent.SetCAL(agent.CreateCAL(cali));
        ⋮
int text_id;
text_id = agent.TextToData(JAPANESE,               (d)
                "発話したいテキスト");
agent.DataToAudio(text_id);
        ⋮
double para[5];
agent.CalcParameter(para);                         (e)
```

図 1: 本ライブラリの使用例

5 むすび

本稿では，多言語の TTS に対応した Lip Sync ライブラリの作成を行った．

今回は日本語の TTS として MS Windows Speech API4.0(WSAPI) に対応した L&H TruVoice を，英語の TTS として TruVoice とエジンバラ大学の Festival Speech Synthesis System を使用した．今後は，他の TTS を調査しそれらの機能の違いに対応できるよう拡張し，さらに現在は同時に 2 人の発話しかできないので，多人数のエージェントへの対応を実現したい．

参考文献

[1] Cohen and Massaro, "Modeling Coarticulation in Synthetic Visual Speech", Thalmann and Thalmann (eds.), *Models and Techniques in Computer Animation*, Springer, 1993, 139–156.