機械学習によるリップシンクアニメーション自動生成技術とFINAL FANTASY VII REMAKEのアセットを訓練データとした実装実例

中田 聖人，Gracia Gil Leandro, 原 龍，岩澤 晃
（株式会社スクウェア・エニックス）

はじめに

本講演は写真撮影、およびSNS投稿可能です

講演資料は後日、CEDiLに公開されます

注意事項

本セッションで紹介する技術は
セッション講演当日(2022年8月25日)現在、
リリース済みの作品に実装された技術を扱っていません。

開発中の内容を扱っており、
将来における当該技術を用いた作品の公表ならびに
発売を意図するものではありません。

講演者紹介

中田 聖人

株式会社スクウェア・エニックス
テクノロジー推進部
R&Dエンジニア

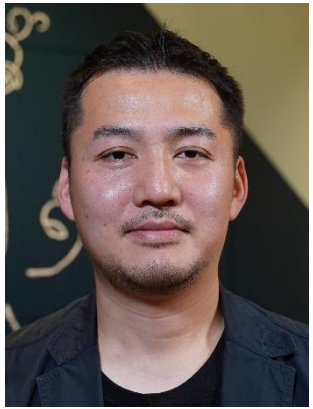
Gracia Gil Leandro

株式会社スクウェア・エニックス
AI部
シニアAIエキスパート

講演者紹介 (cont'd)

原 龍

株式会社スクウェア・エニックス
第一開発事業本部 ディビジョン１
リードアニメーションプログラマー

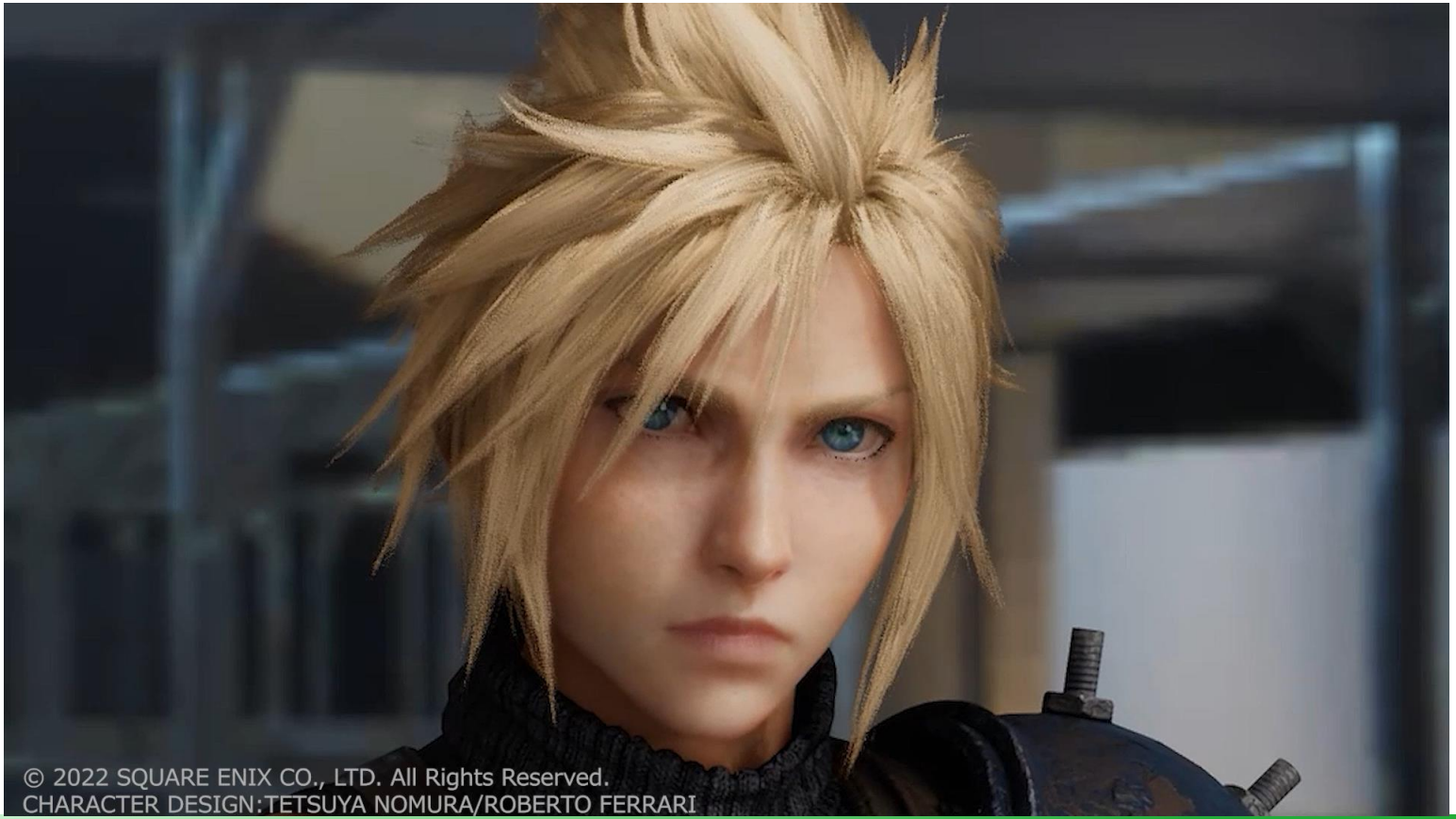


岩澤 晃

株式会社スクウェア・エニックス
第一開発事業本部 ディビジョン１
フェイシャルディレクター

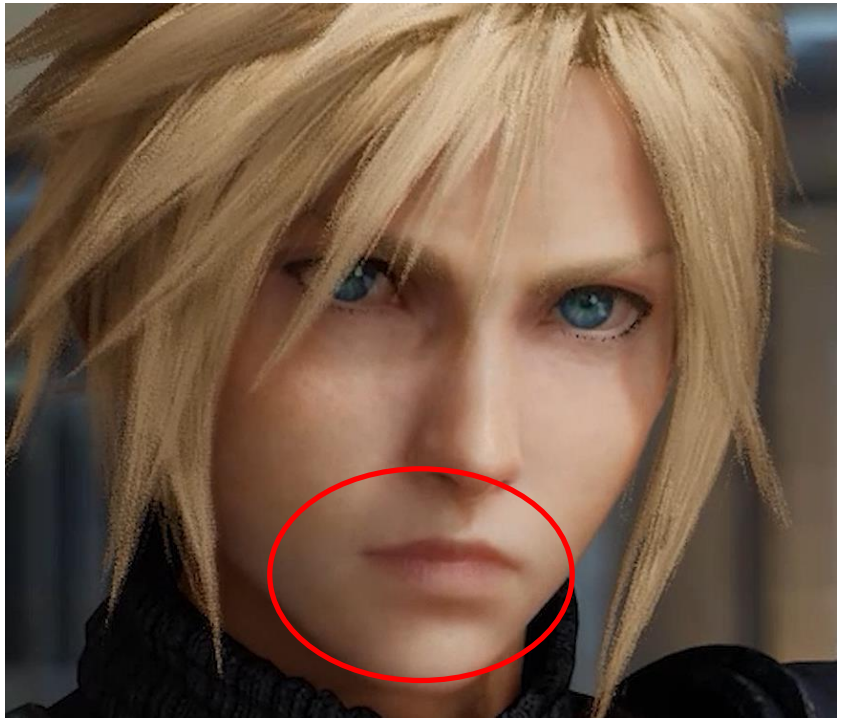
突然ですが、
次の2つの動画を見比べてください

動画①

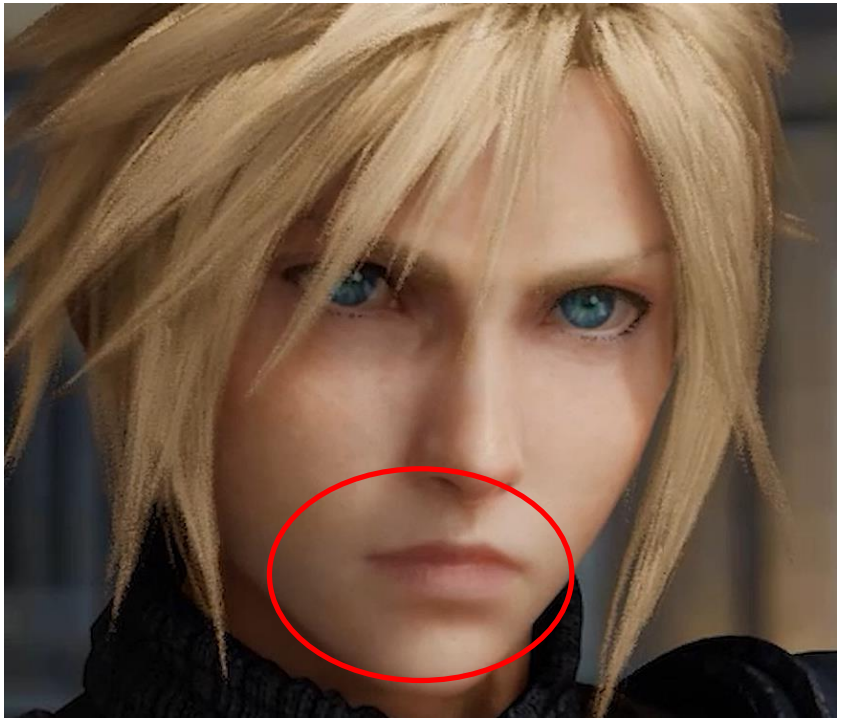


動画②

リップシンクアニメーション



動画①：音声と合わない口パク



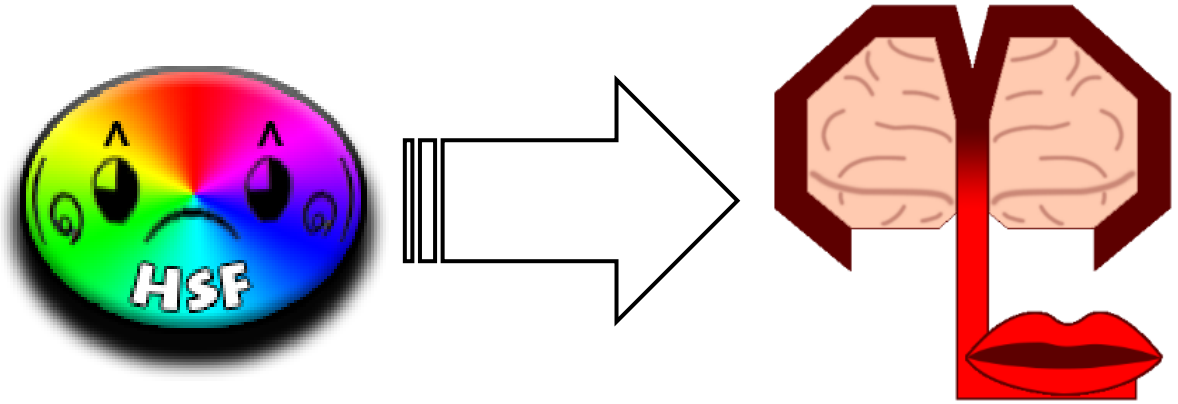
動画②：リップシンクアニメーション

とはいっても

プロダクト内の音声は数時間～数十時間と膨大である

全ての音声に対し、リップシンクアニメーションを
手作業で作成するのは大変な作業である

弊社では



HappySadFace (HSF)

音素解析ベース

Lip-Sync ML

機械学習ベース

参考：
『FINAL FANTASY VII REMAKE』における
キャラクターアニメーション技術 (CEDEC 2020)
https://cedil.cesa.or.jp/cedil_sessions/view/2304

講演内容

- 前提知識のおさらい
- HappySadFaceとLip-Sync ML
- システムの構成、その他の機能

中田

- 訓練データ
- 機械学習モデル
- 使用された機械学習技術

Leandro

研究開発

- Lip-Sync MLの運用事例
- Lip-Sync MLに対応した
  アセットパイプラインの紹介

原、岩澤

プロジェクト

講演では割愛する内容

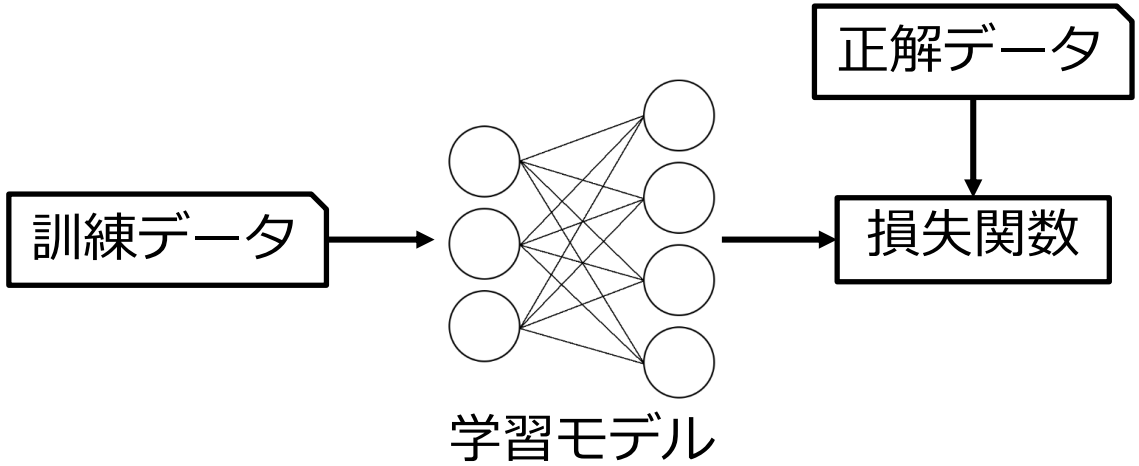
- HSFに関する品質向上のための処理
- 一部の機械学習の詳細
  - 畳み込みネットワークやResNetなどの詳細
  - Transformer encoderやAttention機構の詳細
- プロジェクトで行う一部の処理の詳細
  - 感情解析処理の詳細

得られる知見

- 機械学習とMayaツールの連携方法
- 音声やアニメーションに関する機械学習技術
- 機械学習といった新しい技術を
  プロジェクトに導入するために必要だった実装

前提知識のおさらい

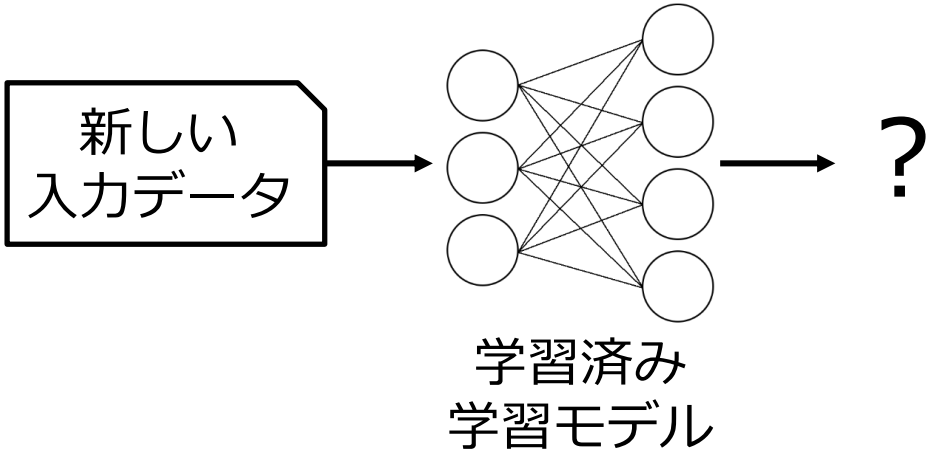
前提知識：機械学習



教師あり学習

… 訓練データを入力し、損失関数が小さくなるように、
学習モデル内のパラメタを修正する処理

前提知識：機械学習 (cont'd)



推論

… 新しい入力データを与え、学習をもとに結果を出力する処理

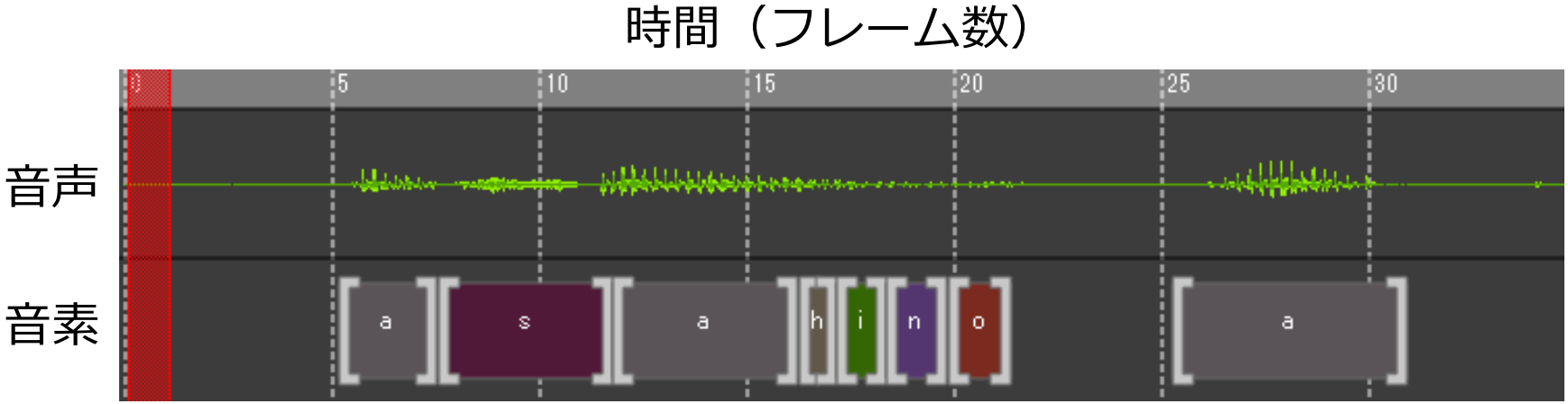
「推論」という単語は、中田のパートでも少し出てくる

前提知識：音素

HSFでは、個々の母音や子音といった
その言語の音として分解可能な最小単位として定義

日本語ではローマ字表記、英語では発音記号表記の
1単位と考えてもらえればよい

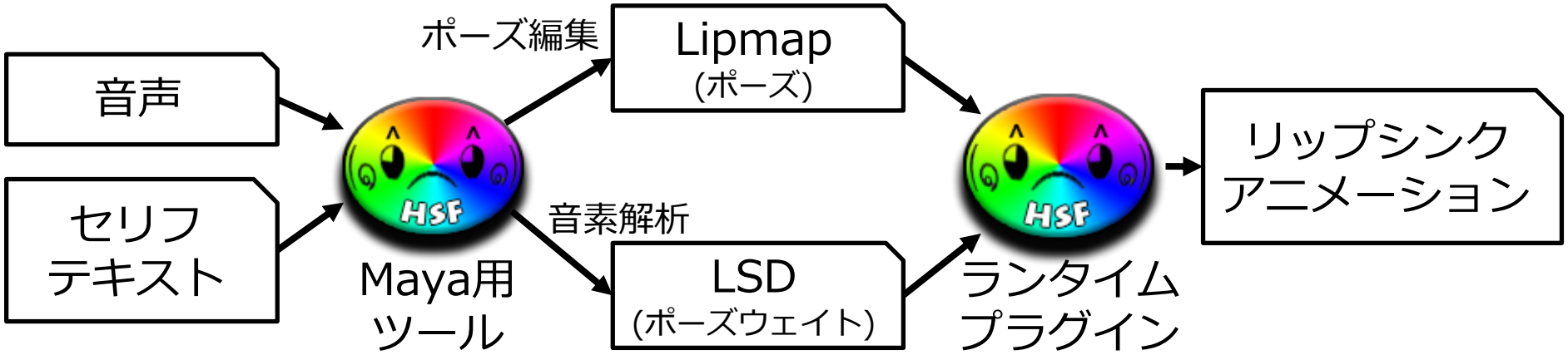
前提知識：音素解析



音声のどの部分がどの音素に対応するかを解析

HappySadFaceとLip-Sync ML

HappySadFace (HSF)



音声とセリフテキストから音素解析を行い、
その結果からアニメーションを自動生成

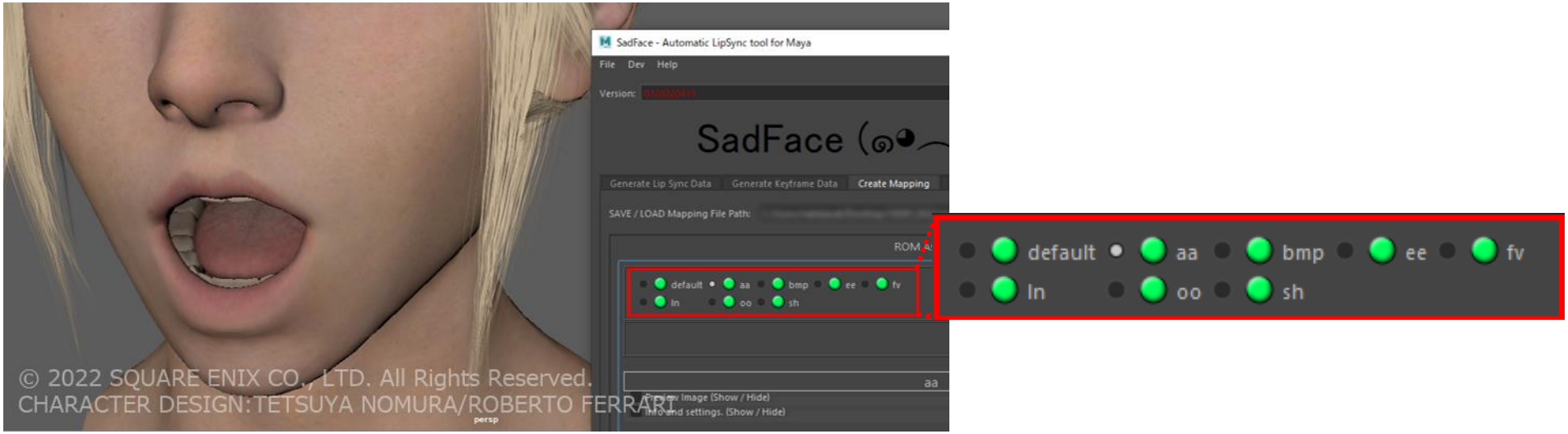
HSFでやっていること

解析された各音素に口の形を割り当てている。

1つ1つの音素に口の形そのものを登録するのは非効率

- 音素の種類は多数ある（日本語：40, 英語：42）
- 異なる音素が同じ口形状になるケースがいくつかある

Lipmap



リップシンクアニメーション生成に必要である
いくつかの口のポーズ（形状）を記録しているデータ

Lipmapのポーズ例：default

基準となる、口を動かさないポーズ

Lipmapのポーズ例：aa

口を大きく開いたポーズ

Lipmapのポーズ例：ee

口を横に広げたポーズ

ポーズのブレンド

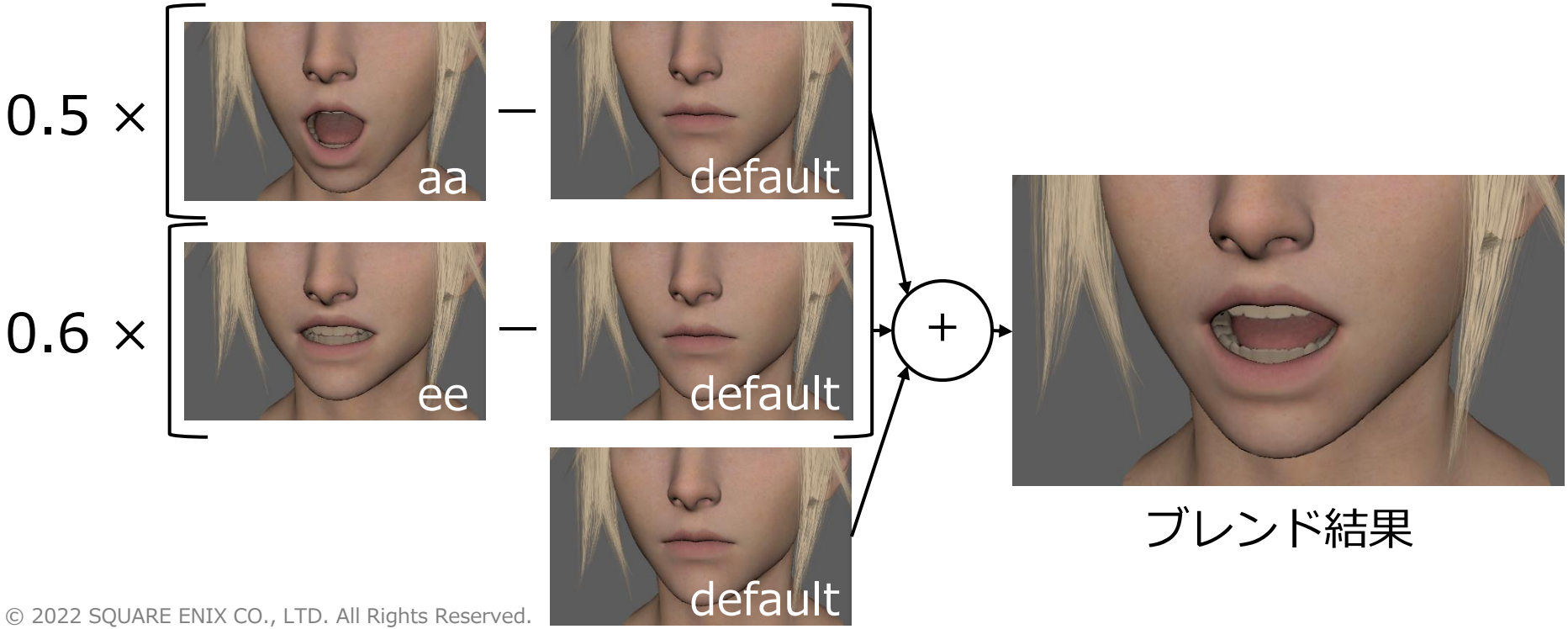
ポーズウェイトをもとに、
defaultポーズとの差分をブレンド

ポーズウェイト

… どの程度、そのポーズを反映させるかを示す数値

ポーズのブレンド (cont'd)

例：aaポーズのウェイトが0.5, eeポーズのウェイトが0.6の場合



ブレンド結果

Lipmapと音素との対応

音素解析結果

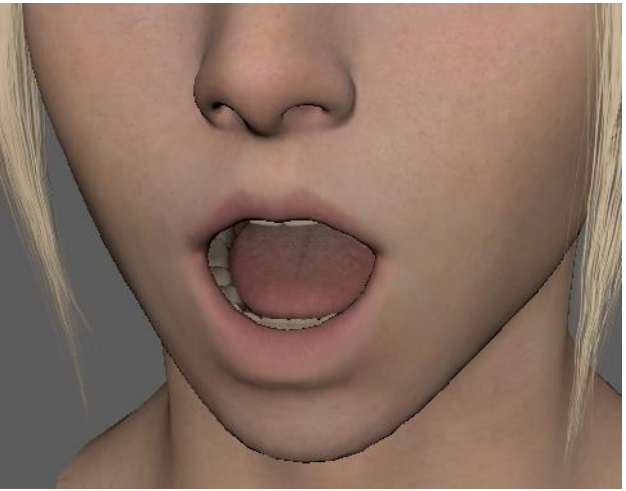
音素表

LSD内の
音素データ

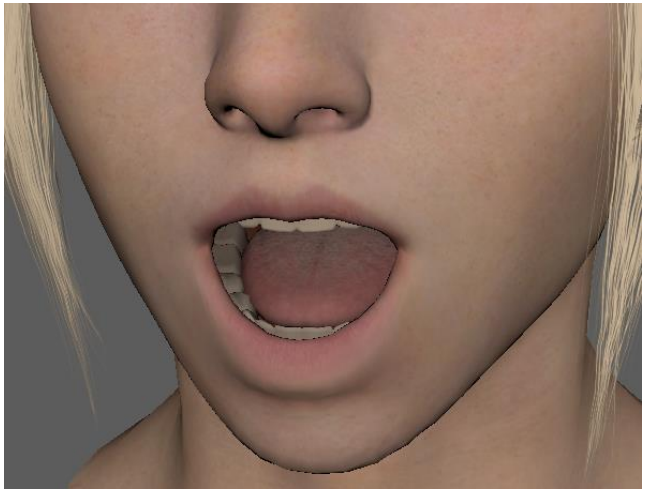
LSD

… アニメーション生成に使用する、
音素のタイミングとその音素でのポーズウェイトを格納したデータ

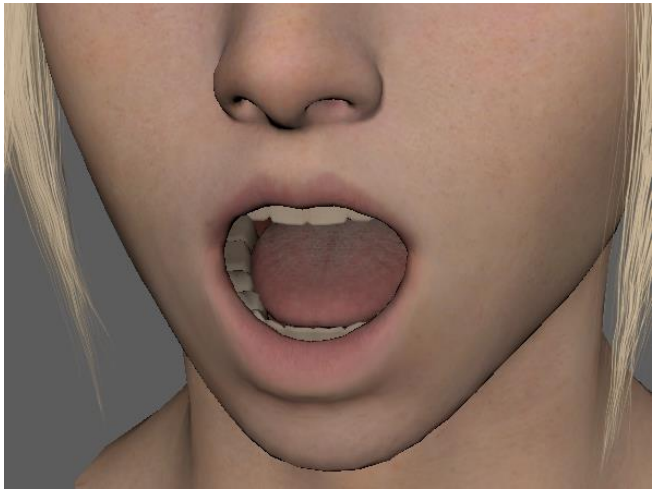
感情ごとのLipmap



デフォルトポーズ用



笑顔ポーズ用



怒りポーズ用

FINAL FANTASY VII REMAKEでは
感情に応じて使用するLipmapを切り替えていた

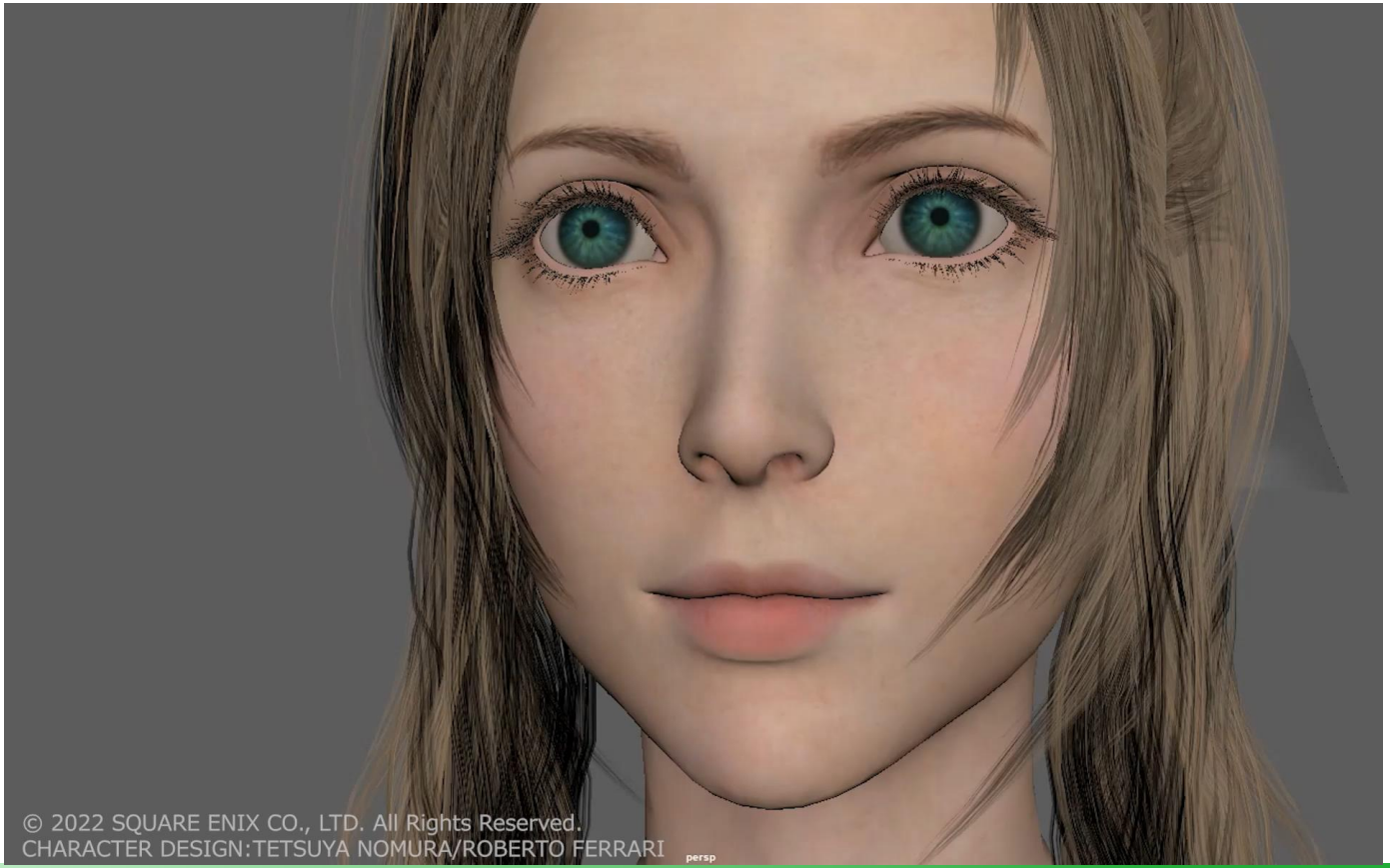
HSFの課題

セリフテキストと音声の不一致に起因する課題があった

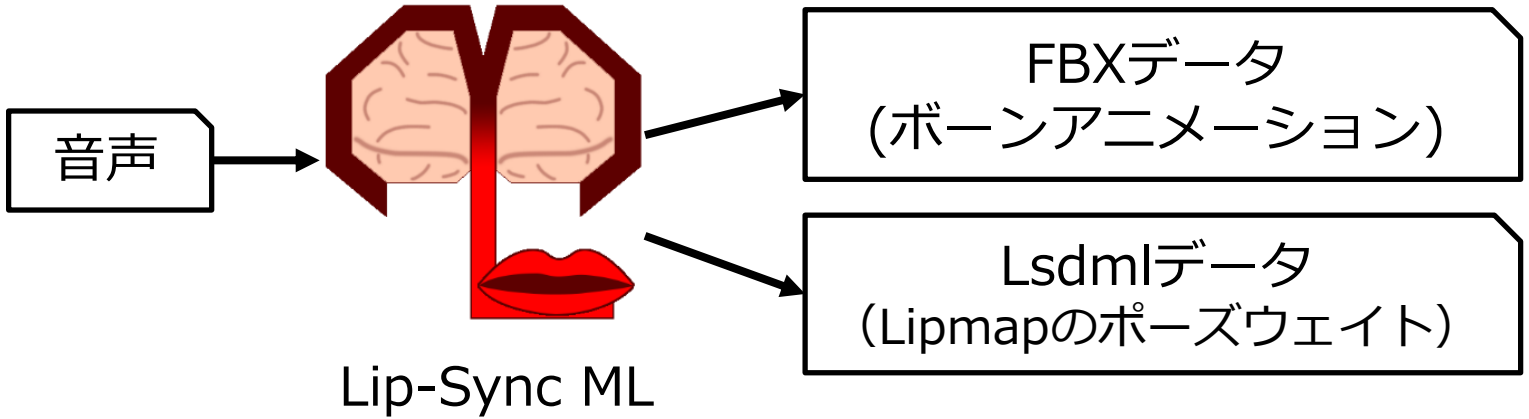
- セリフにない呼吸音に音素が割り当てられ、ずれてしまう
- セリフにないアドリブボイスがあると、精度が落ちる

など

HSFの課題 (cont'd)



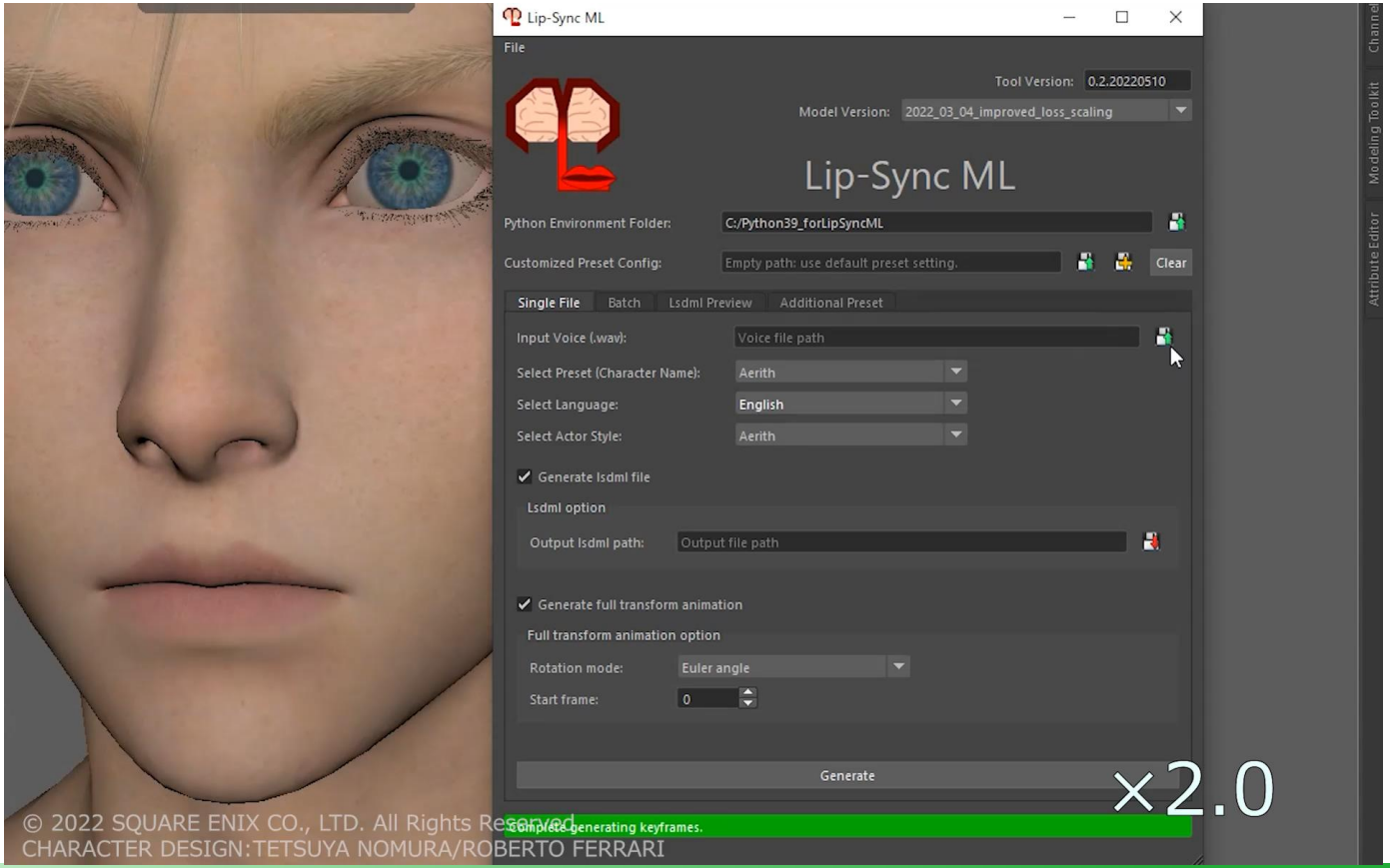
今回開発したシステム：Lip-Sync ML



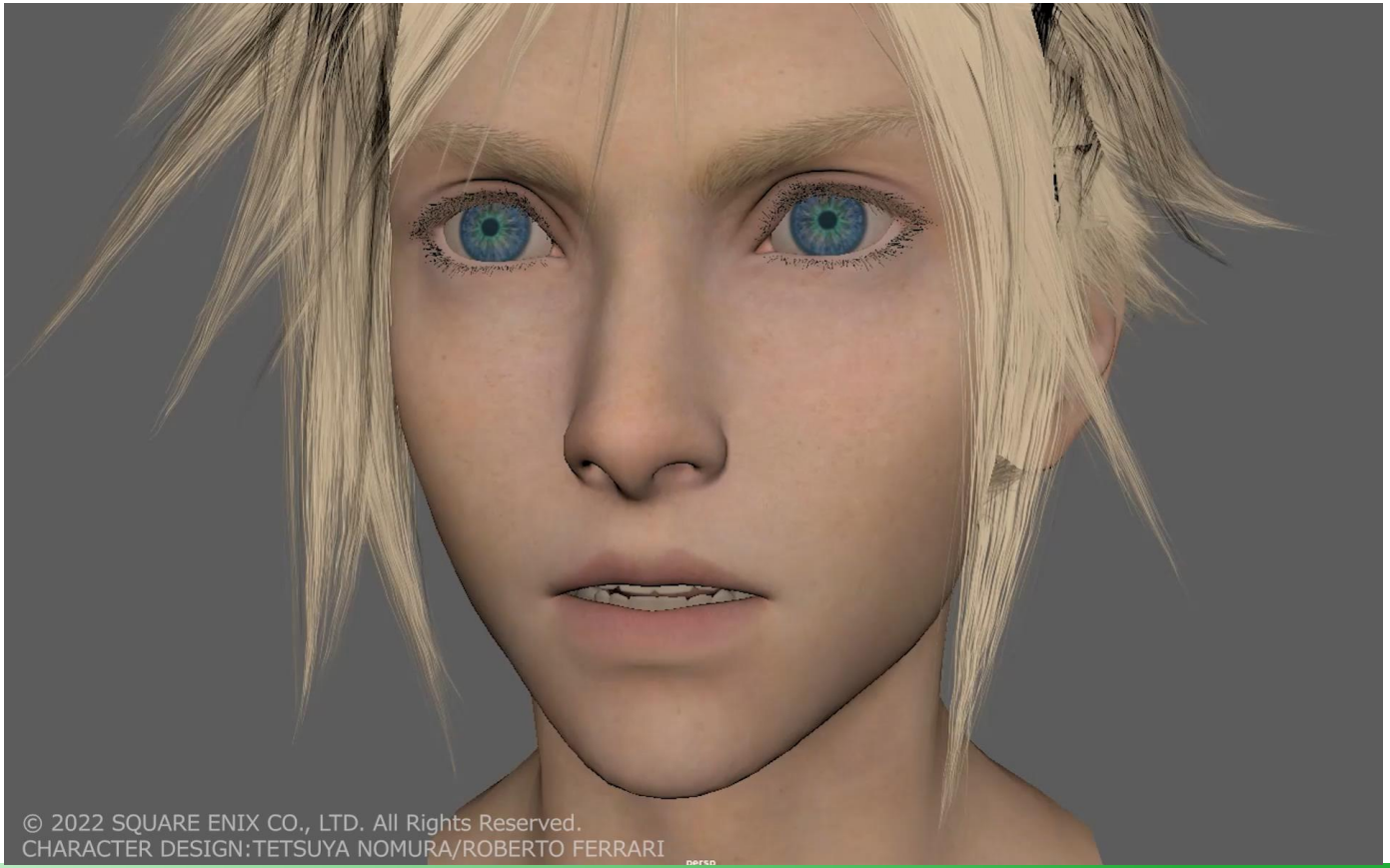
機械学習技術を用いて、音声のみの入力から
直接アニメーションを自動生成するシステム

Lipmapのポーズウェイトとして出力する仕組みもある

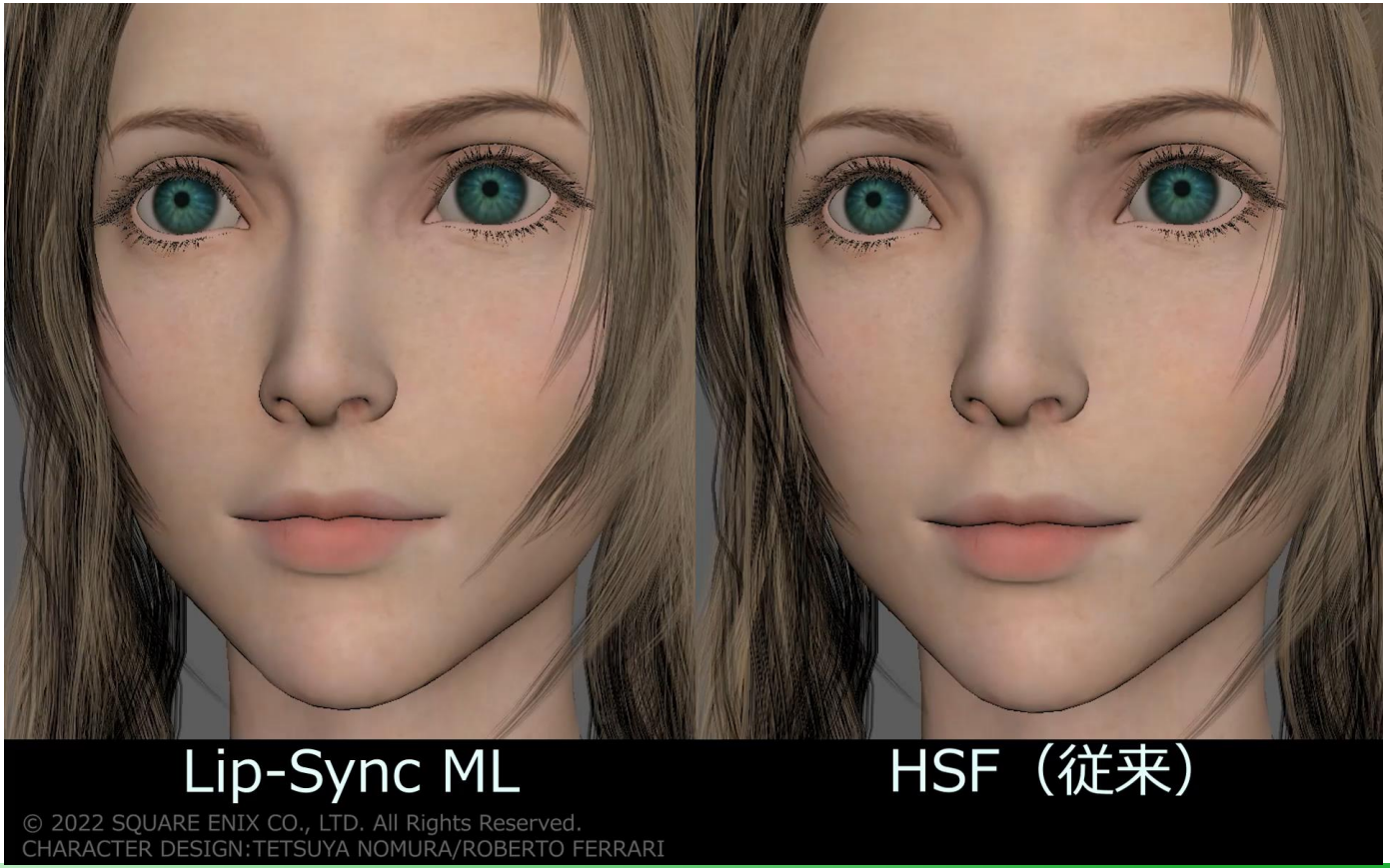
Lip-Sync MLで自動生成する様子



Lip-Sync MLの自動生成結果



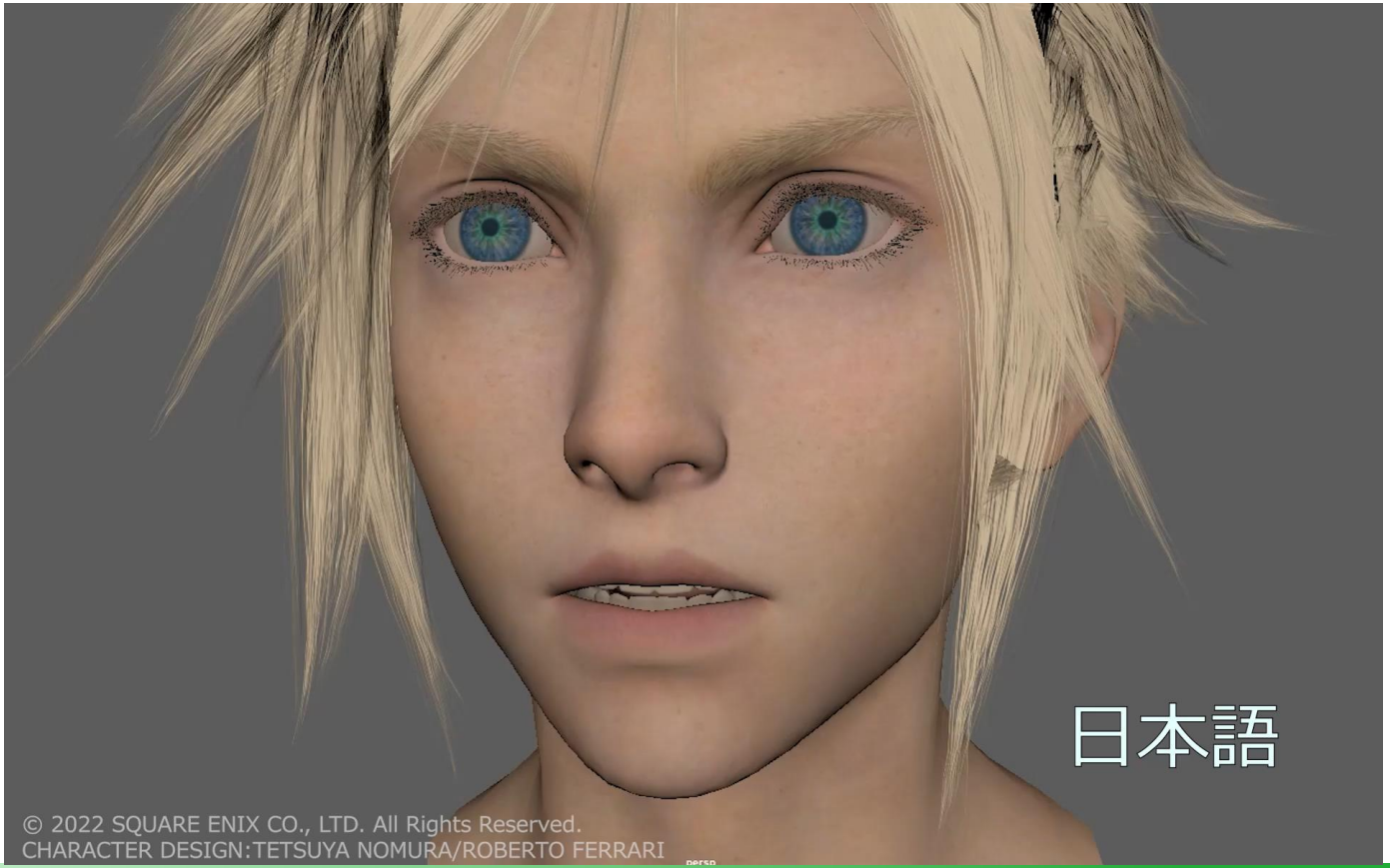
Lip-Sync MLとHSFとの比較（息つぎ）



Lip-Sync MLとHSFとの比較表

項目	Lip-Sync ML	HSF
使用技術	機械学習	音素解析
入力	音声	音声、セリフテキスト
事前準備	訓練データの収集	セリフテキストの準備
クオリティ	高	低
呼吸音などの対応	○	×
対応言語	日本語、英語、 ドイツ語、フランス語	日本語、英語、 ドイツ語、フランス語
編集難易度	難	易

他言語での結果



人型以外のモデルでの結果

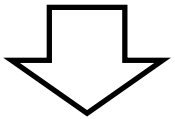
システム構成

Mayaと機械学習との連携

Maya 2020以前… Python 2.7

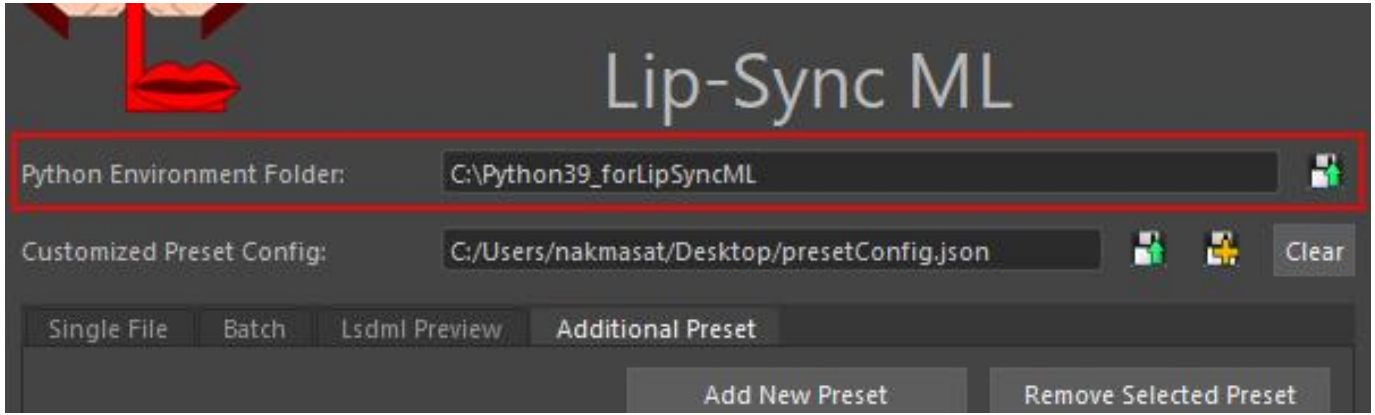
機械学習の推論システム… Python 3.6~3.9
(Tensorflowなどを使用)

Maya上のPythonでは、推論システムを動かせない



- 推論システム用の**Python3仮想環境**をvenvで作成し、Mayaから別プロセスとして呼び出す
- データのやりとりは**jsonファイル**を通して行う

仮想環境のメリット



Lip-Sync ML開発側で
必要なPythonモジュールをあらかじめ準備できる

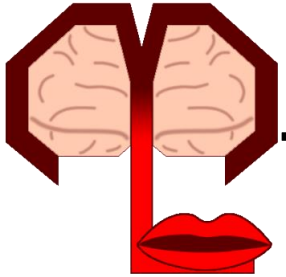
- プロジェクト側ではPythonモジュールのインストールは不要
- Python環境、バージョンの違いで発生する不具合も減らせる

jsonファイルとしてやりとりするデータ

中間アニメーションデータ

… フレームごとの、各ボーンのトランスフォームを格納したデータ

Lip-Sync MLのシステムの流れ



推論
システム

③出力

Lsdmlデータ

中間アニメーション
データ (json)

②別プロセス
として呼び出し

④中間アニメーションデータを
Maya上に適用

音声

①入力

Maya用
ツール

リップシンクアニメーション
(fbx)

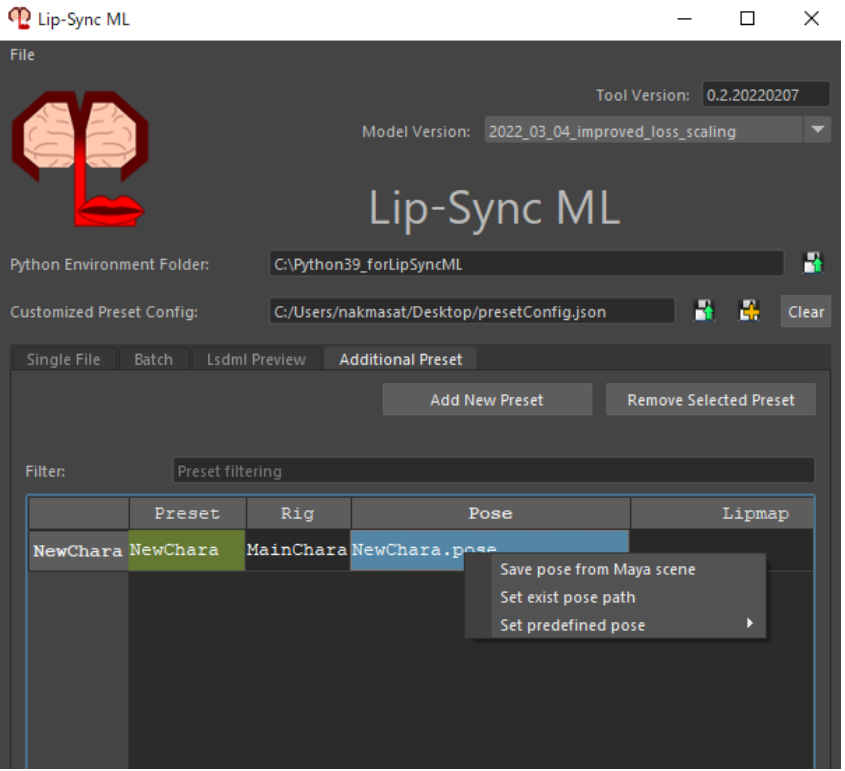
バッチ処理

	A	B	C	D	E	F	G	H	I	J	K
1	Audio	Preset	Language	Actor	Lsdml	Rig	Maya	FBX	FBXRoot	RotationMode	
2	jpn¥EV_M	Cloud	Japanese	Cloud	jpn¥test.ls	D:¥00_wo	jpn¥testR	jpn¥testR	C_Neck_a	0	
3	eng¥EV_M	Cloud	English	Cloud	eng¥test.l	D:¥00_wo	eng¥testR	eng¥testR	C_Neck_a	1	
4	fra¥EV_M	Cloud	English	Cloud	fra¥test.js	D:¥00_wo	fra¥testRe	fra¥testRe	C_Neck_a	0	
5	ger¥EV_M	Cloud	English	Cloud	ger¥test ls	D:¥00_wo	ger¥testR	ger¥testR	C_Neck_a	1	
6											

設定用csvファイルを入力し、
コマンドラインから複数の音声を対象に
Lip-Sync MLの処理を実行する仕組み（大量生産向け）

その他の機能

新規キャラクタへの対応

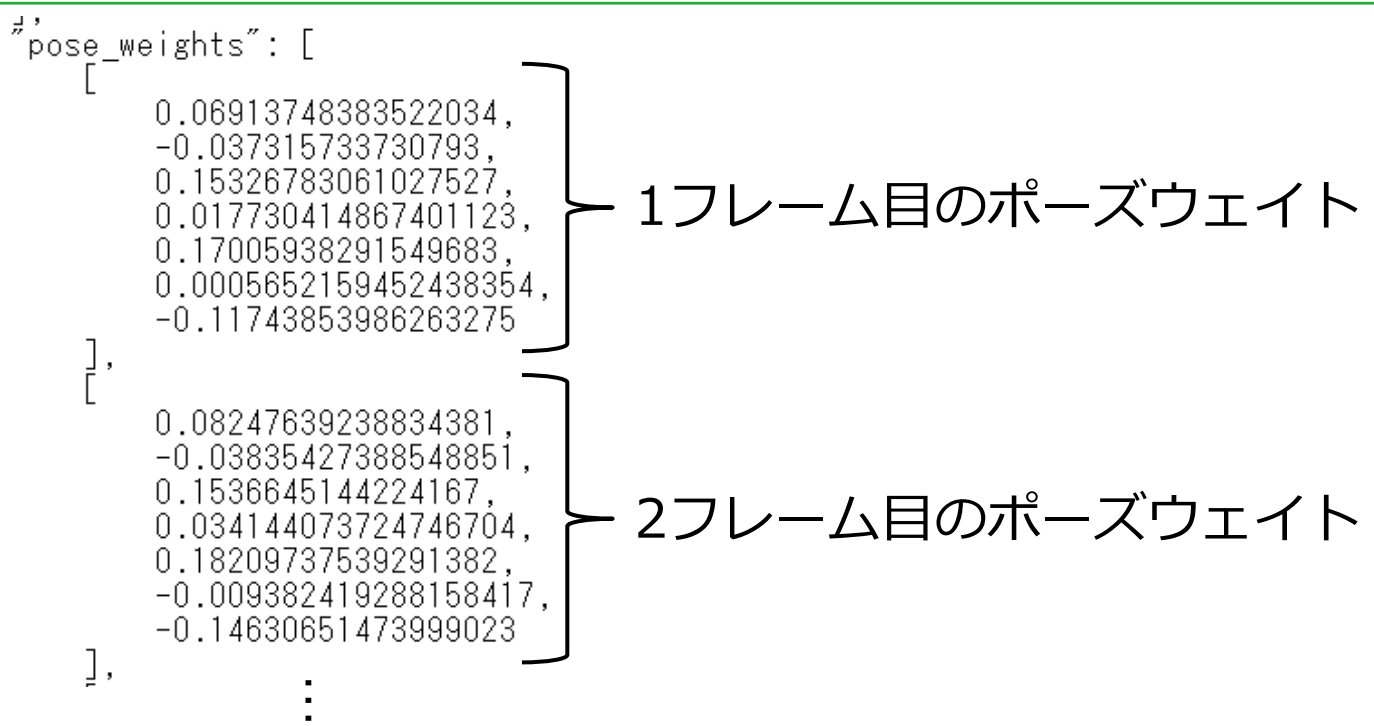


新規キャラクタの
バインドポーズ（初期ポーズ）を
登録することで対応

※ボーン構造が学習モデル内のもの
と一致する必要がある。

訓練データにないキャラクタの結果

Lsdml



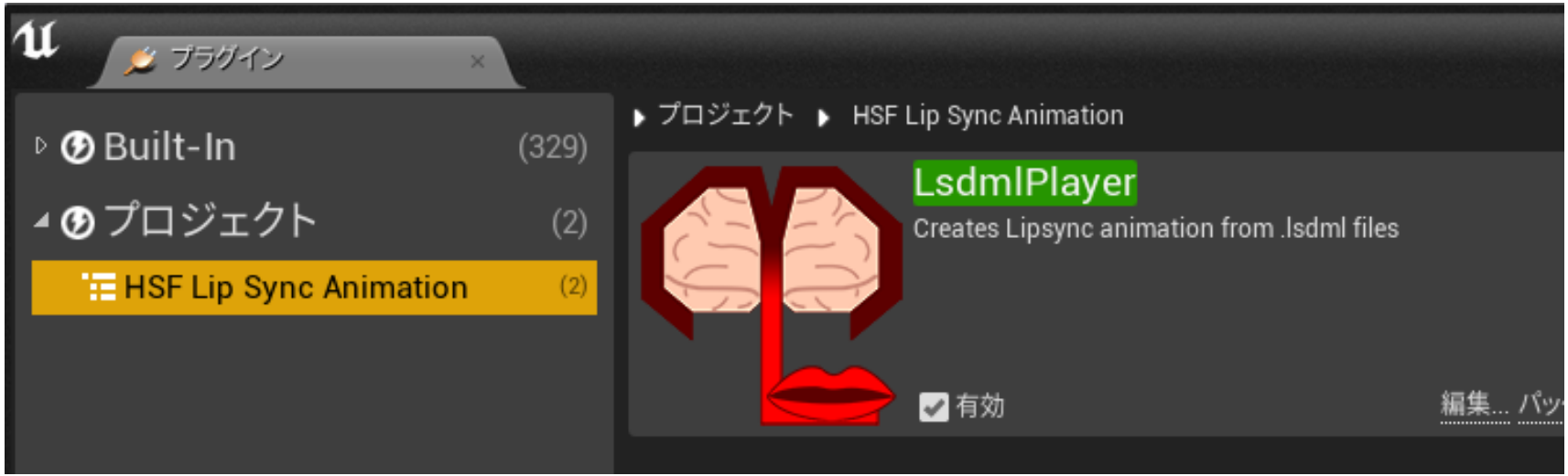
毎フレームのLipmapのポーズウェイトを格納したデータ

Lsdml (cont'd)

- HSFに近いワークフローを実現できる
- Lipmapが設定されれば、**別の骨構造のモデル**にも
  アニメーションを適用できる

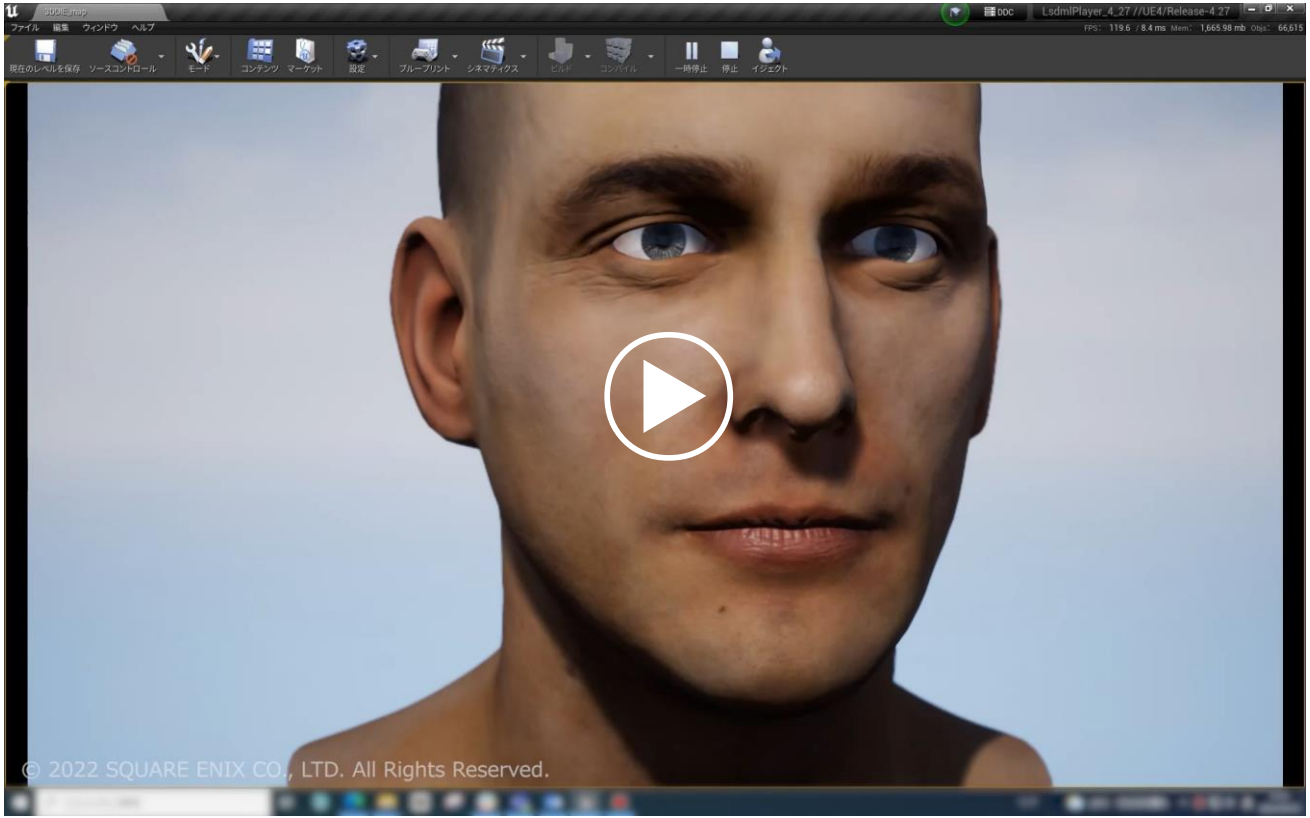
- ボーンアニメーションでの出力に比べ、
  少しクオリティが下がる

LsdmlPlayer



Lsdmlを再生するためのUnreal Engine用プラグイン

LsdmlをUnreal Engineで再生した例



モデル：Facial Rig based on FACS (https://www.turbosquid.com/3d-models/3d-model-of-rig-based-facs/1005479)

ツールとしての課題

- 生成アニメーションの編集手段
  … FBX, Lsdmlともに毎フレームごとに格納されたデータである
- アニメーションの後処理
  … 最後必ず口を閉じるようにするなど
- 生成結果の妥当性を自動で確認する手段
- 感情を反映したアニメーションの生成手段

機械学習によるリップシンクアニメーション自動生成技術と FINAL FANTASY VII REMAKEのアセットを訓練データとした実装実例

機械学習の詳細

Graciá Gil, Leandro

訓練データ

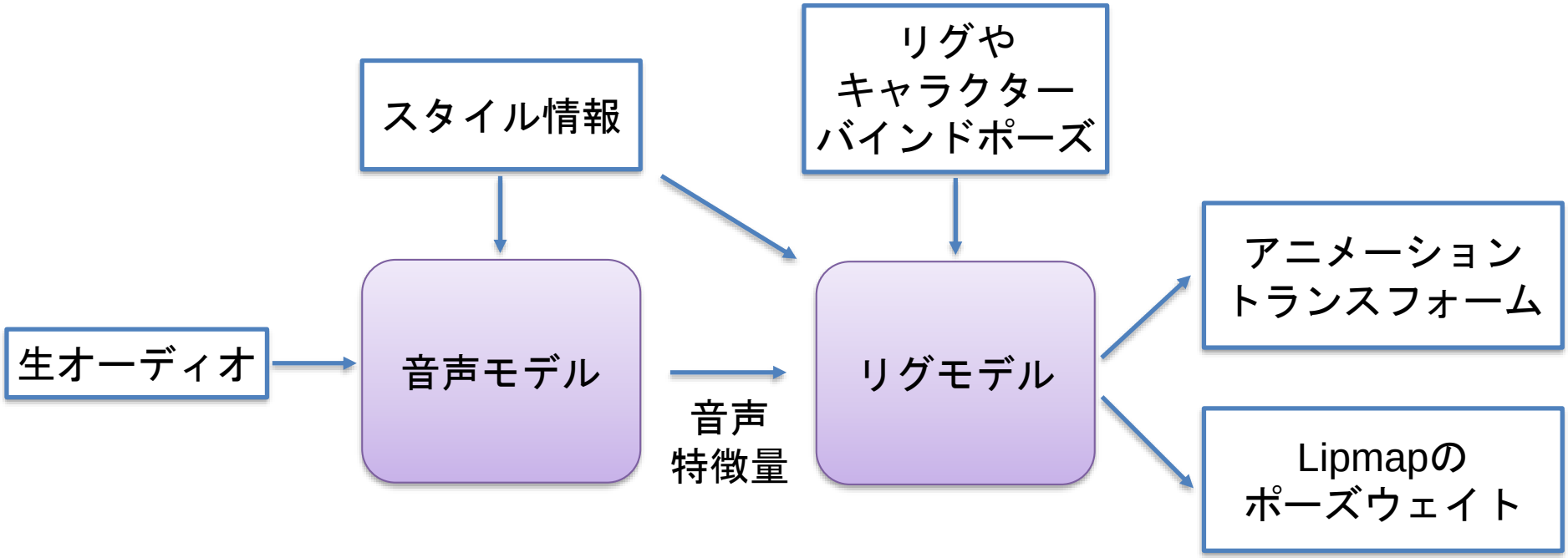
- カットシーンから抽出された音声とそれに同期したアニメーション
- データ量: 約３時間半
- キャラクター数: 53体, リグ: 3種類, 言語: 日英
- 感情ラベルなし
- 多数の短いデータは、同じキャラクターや同じ言語同士で連結

データ拡張

- 目標：音声の速度とピッチの変化に対するロバスト性の向上
- 拡張用クリップの生成
  - オーディオとアニメーションの速度をランダムに変化
  - オーディオのピッチをランダムに変化
- 生成したものは元の学習用データセットに混ぜる

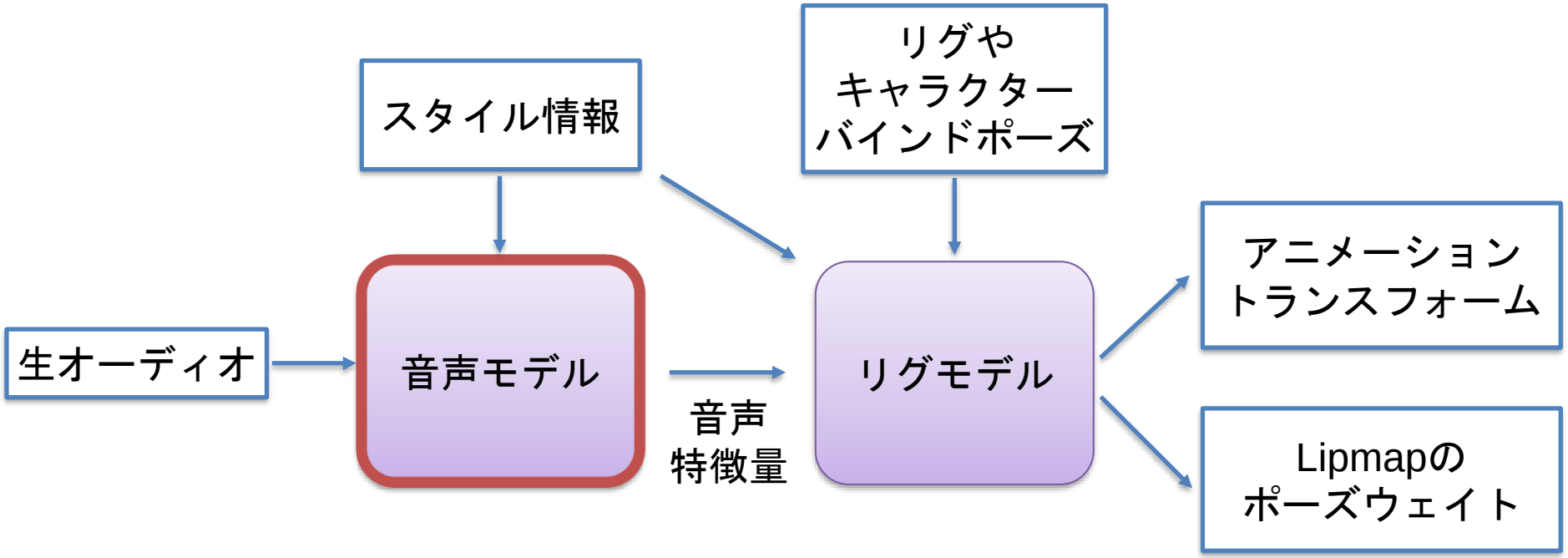
機械学習モデル

- ２つのモデルで学習



機械学習モデル

- ２つのモデルで学習

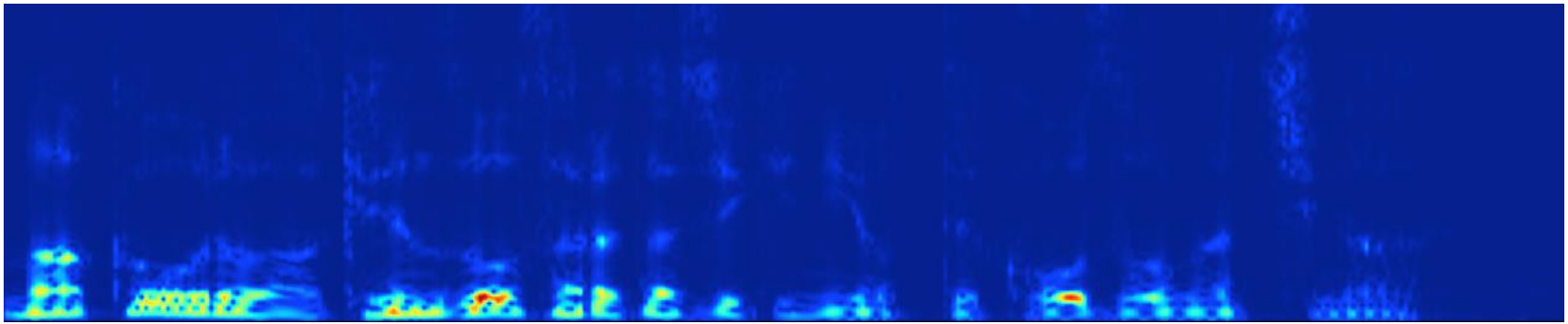


オーディオ処理

- オーディオをモノラルに変換
- 19200 Hz にリサンプリング

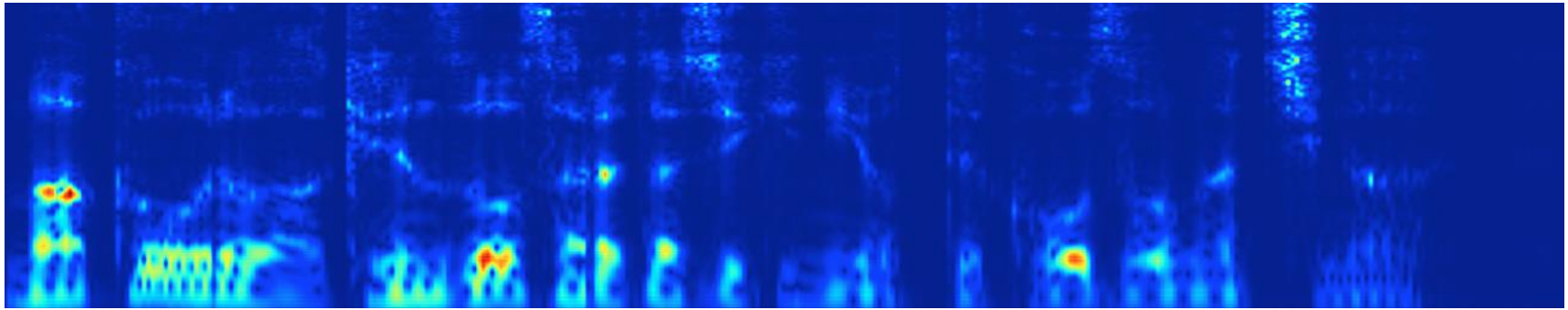
オーディオ処理

- スペクトログラムの計算
  - 窓幅: 200 samples, ストライド幅: 160 samples
  - 出力: 120 Hz



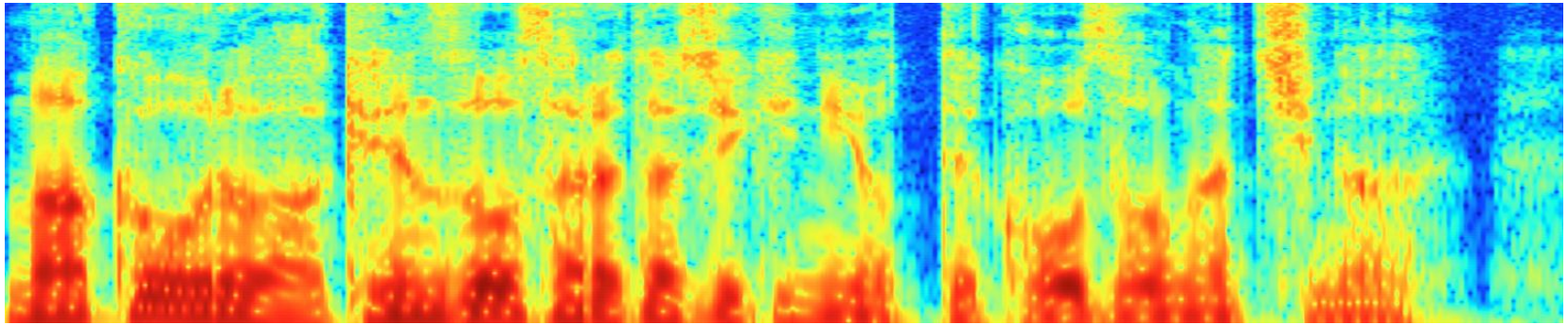
オーディオ処理

- メル尺度に変換 (メル スペクトログラム)
  - 人間の周波数感覚に近い
  - ピッチの変化は縦方向の変化としてあらわされる



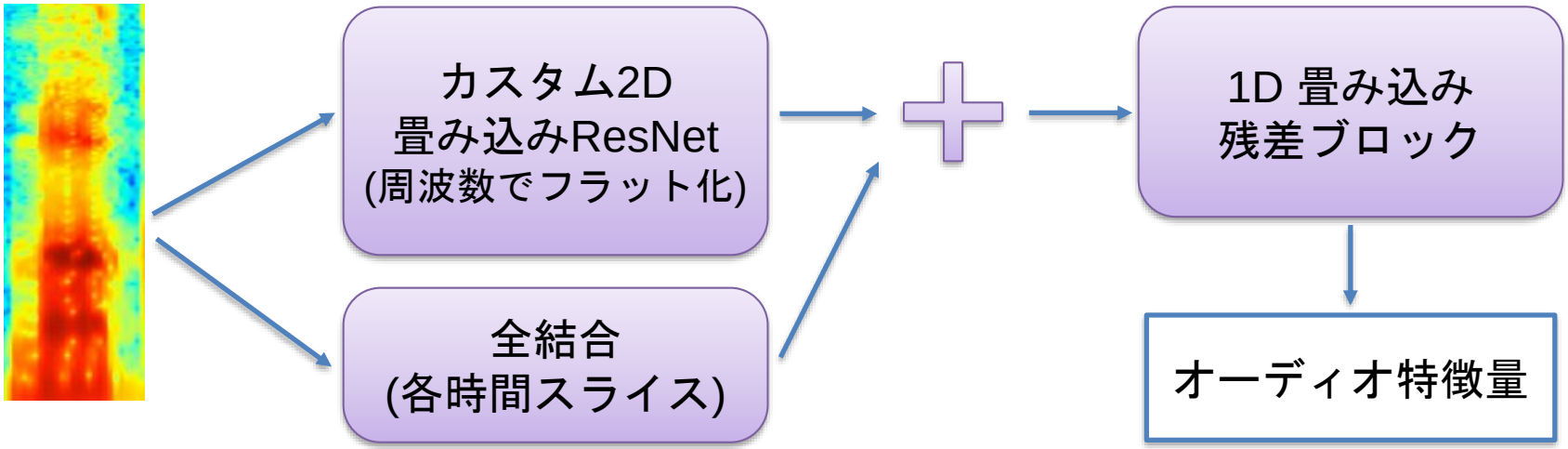
オーディオ処理

- 値の対数を計算 (ログ メル スペクトログラム)
  - 人間の音量感覚に近い



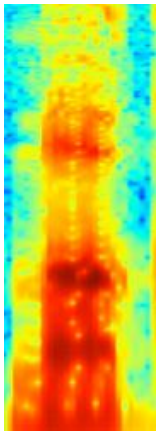
スペクトログラム処理

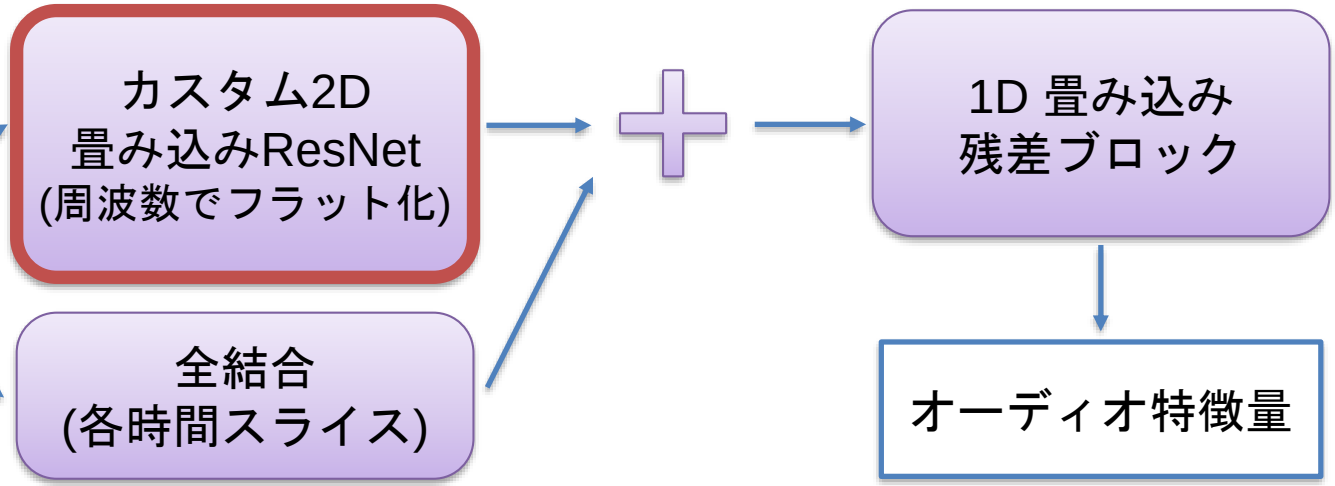
- 相対ピッチ: カスタム2D 畳み込みResNet
- 絶対ピッチ: 全結合レイヤー



スペクトログラム処理

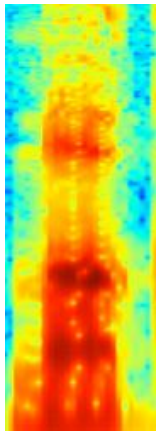
- 相対ピッチ: カスタム2D 畳み込みResNet
- 絶対ピッチ: 全結合レイヤー





カスタム2D畳み込みネットワーク

- 目標：ストライドを使用して周波数情報を深度チャネルに段階的に変換



2D conv
channels=64
kernel=7x7
stride=1x4
ReLU

Batch
norm

2D conv
channels=96
kernel=7x7
stride=1x4
ReLU, no bias

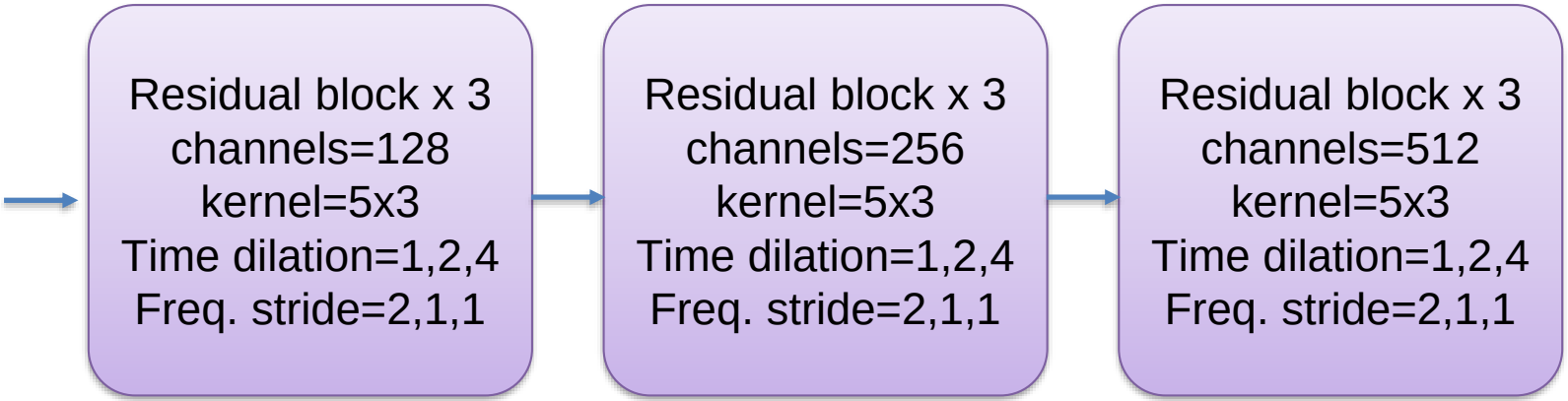
Batch
norm

2D conv
channels=128
kernel=7x7
stride=1x4
ReLU, no bias

Batch
norm

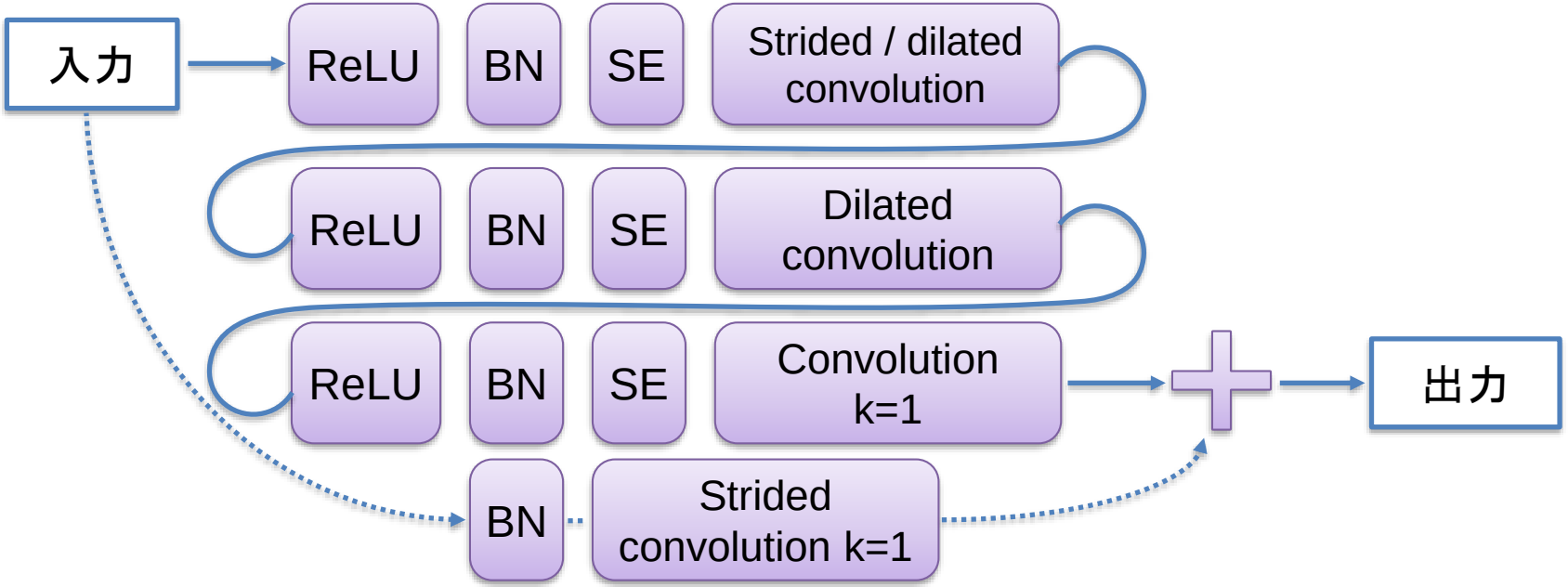
カスタム2D畳み込みネットワーク

- 9つの カスタム dilated residual blocks
- 音声の速度変化への対応力を向上



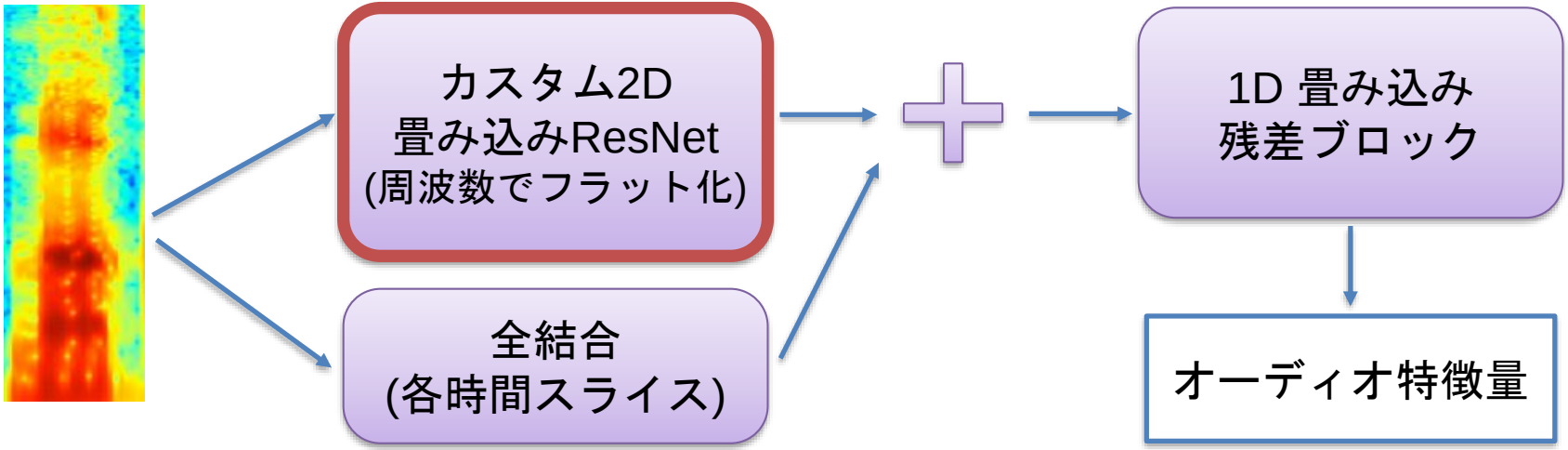
Custom Residual Blocks

- BN: Batch Normalization
  SE: Squeeze-and-Excitation



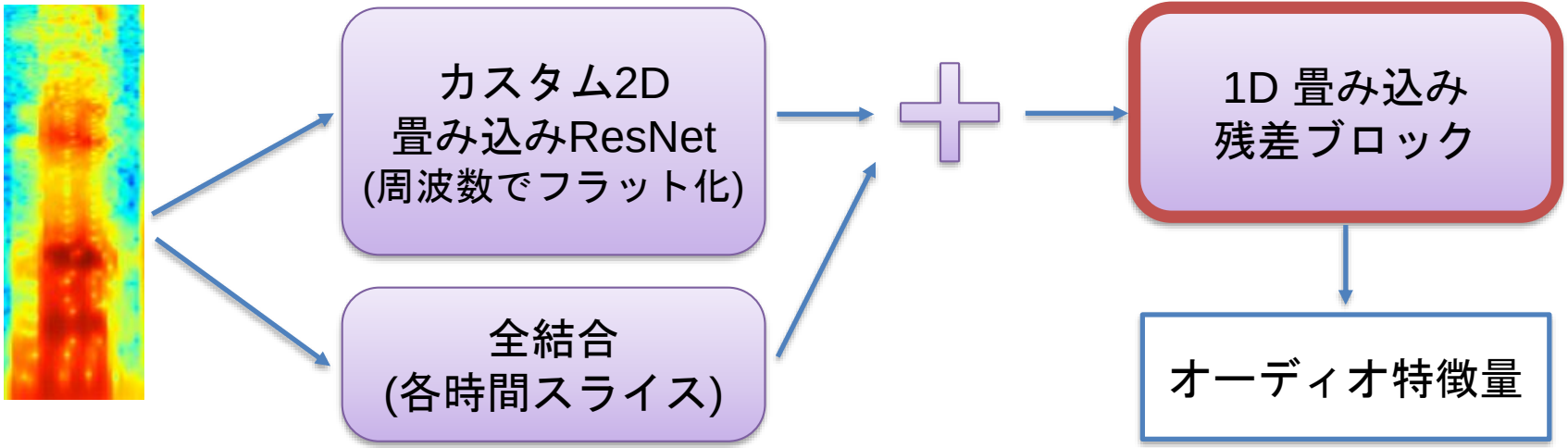
スペクトログラム処理

- 相対ピッチ: カスタム 2D 畳み込みネットワーク
- 絶対ピッチ: 全結合レイヤー



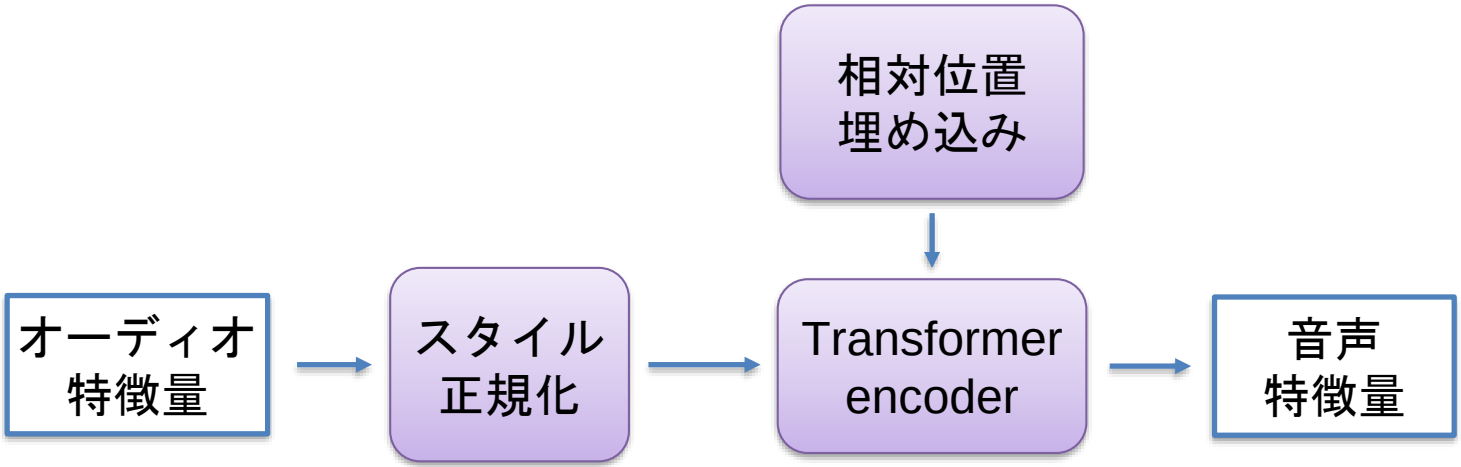
スペクトログラム処理

- 相対ピッチ: カスタム 2D 畳み込みネットワーク
- 絶対ピッチ: 全結合レイヤー



音声特徴量

- オーディオ特徴量を使用した transformer encoder
- 相対位置埋め込み (relative positional embeddings)
- アテンションマスク前後１秒



スタイル正規化

- スタイルは独立な値の集合
  - 言語: 日本語, 英語, …
  - キャラクター: クラウド, ティファ, エアリス, …
  - リグ: 主要キャラクター, モブキャラクター, レッドXIII
- スタイルの値は設定しないことも可能

スタイル正規化

- スタイルごと及びスタイル値ごとに埋め込みを学習
- 選択されたスタイル値の埋め込みとスタイルの埋め込みを加算

言語：なし
キャラ：クラウド
リグ：主要キャラ

スタイル	スタイルの値		
言語	日本語	英語	スペイン語
キャラ	クラウド	エアリス	ティファ
リグ	主要	モブ	レッドXIII

言語

キャラ + クラウド

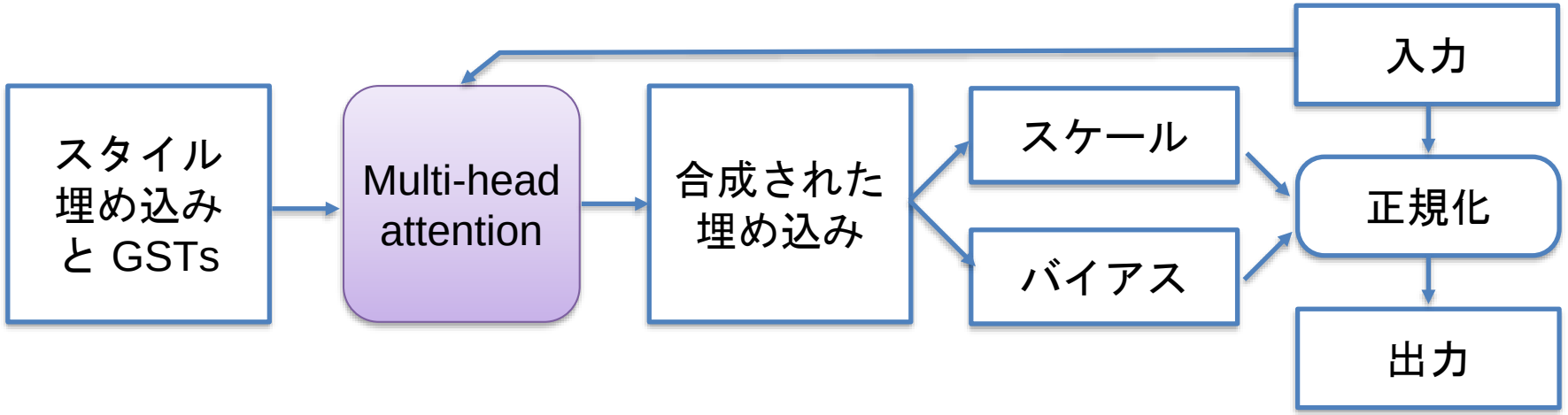
リグ + 主要キャラ

スタイル正規化

- スタイルごとに１つの埋め込み
- 全部のデータで共有される「Global Style Tokens」を追加

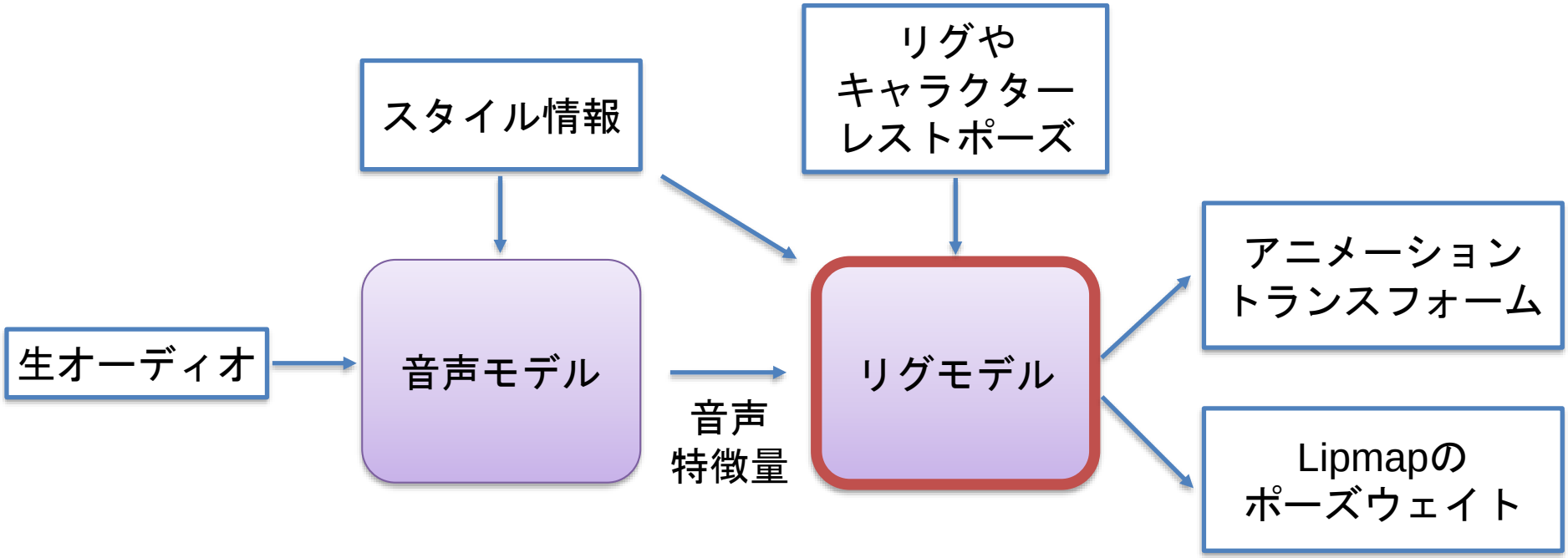
スタイル正規化

- 「Multi-head attention」で埋め込みを合成
- 結果はバイアスとスケールベクトルに分割
- 出力 = 入力 * (スケール + 1.0) + バイアス



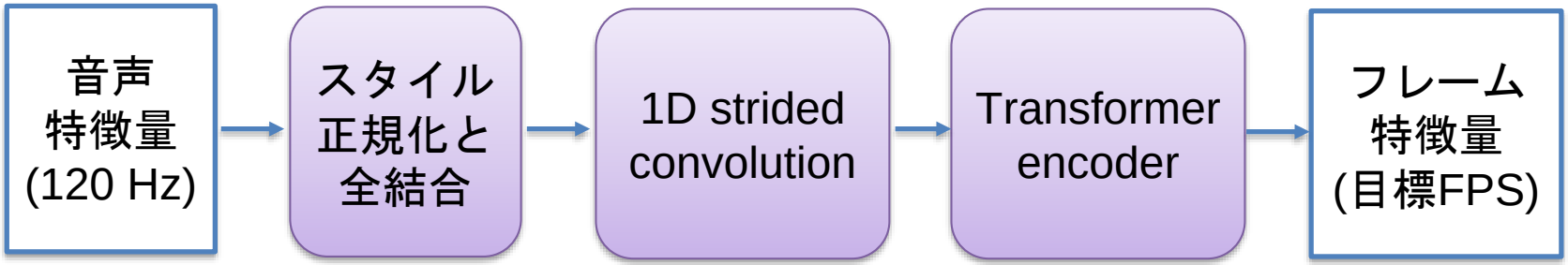
機械学習モデル

- ２つのモデルで学習



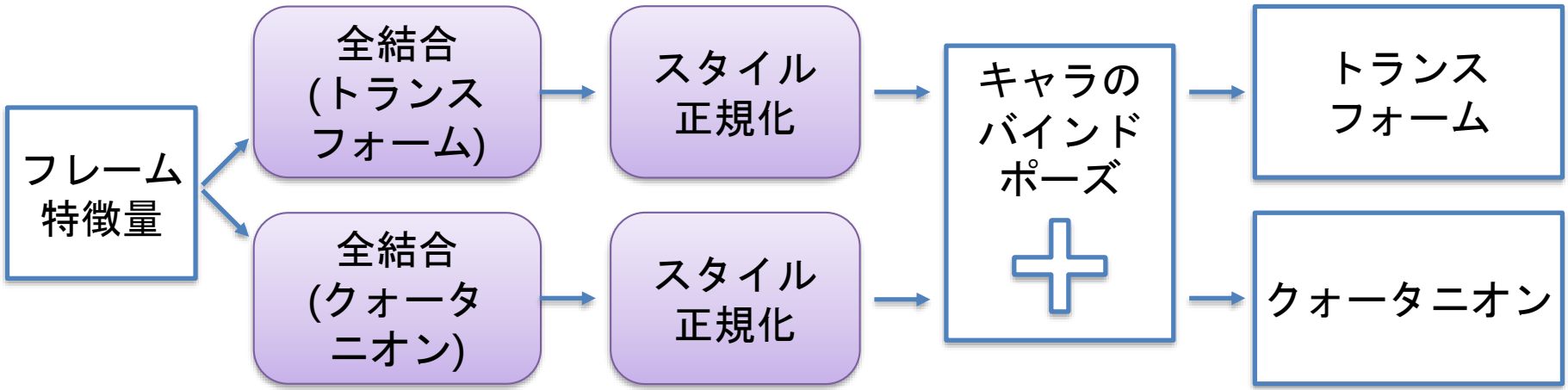
フレーム特徴量

- 音声特徴量は 120 Hz 固定
- 畳み込みのストライドで目標のFPSに変更
- 結果はフレーム特徴量



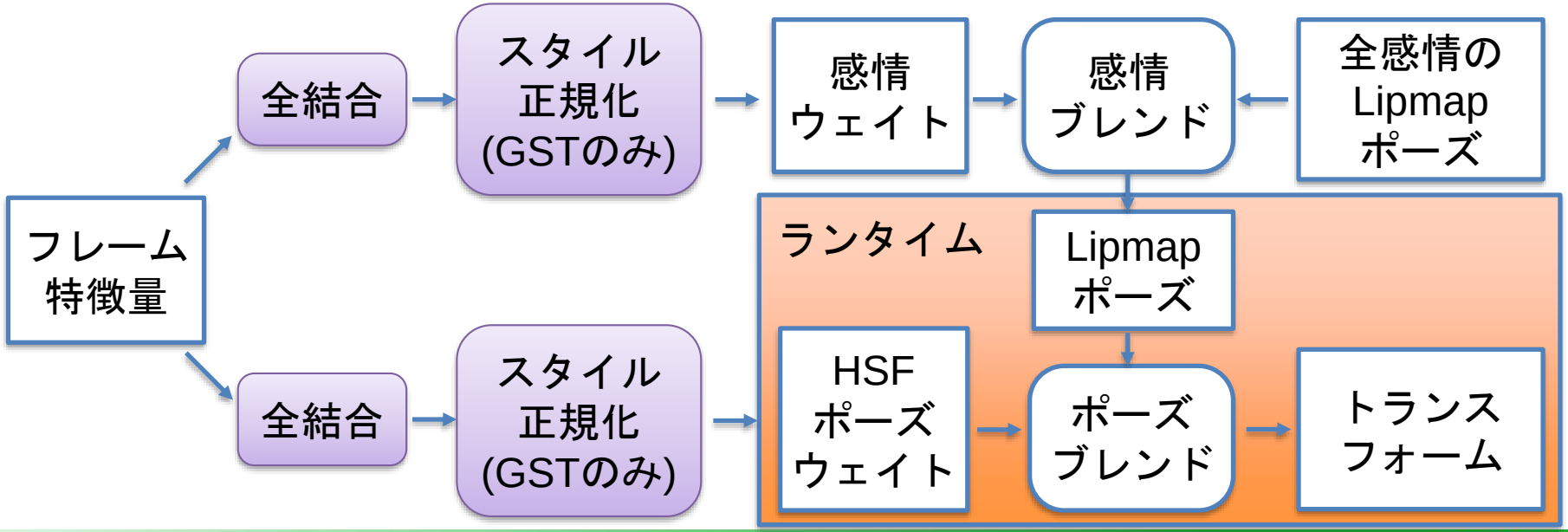
アニメーション・トランスフォーム

- トランスフォーム生成（回転はオイラー角）
- 別途クォータニオン生成
- キャラクターのバインドポーズを結果に加算



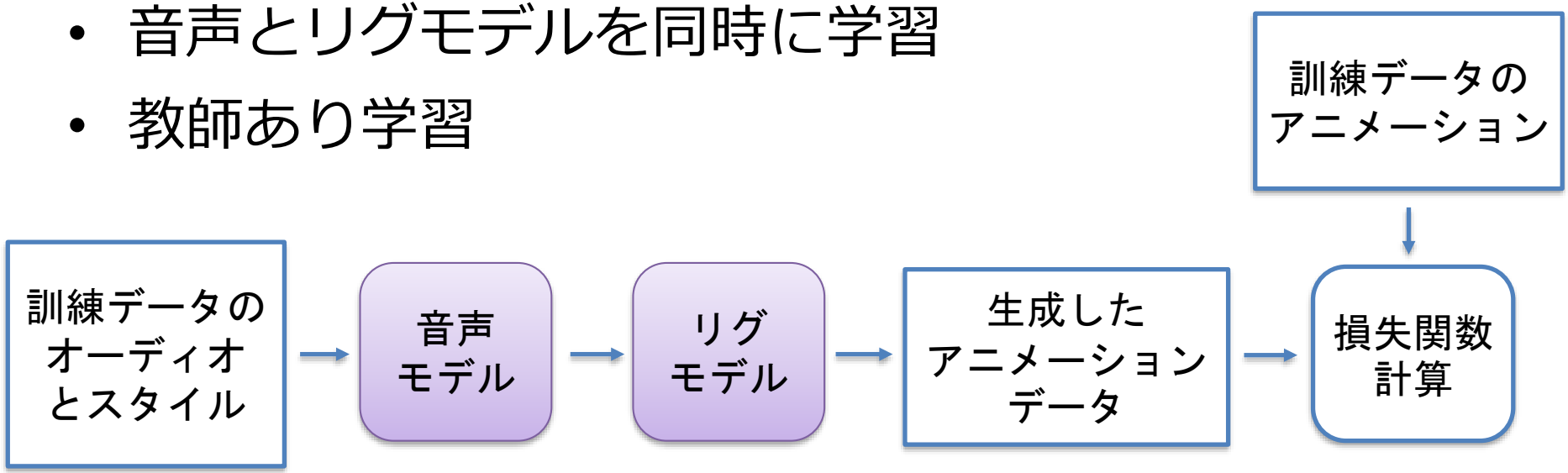
Happy Sad Face Lipmaps

- lipmapでは各感情ごとにポーズセットを持つ
- 全感情ブレンド後、ポーズセットのブレンド



エンド・ツー・エンド学習

- 同期したオーディオとアニメーションを使用
- 音声とリグモデルを同時に学習
- 教師あり学習

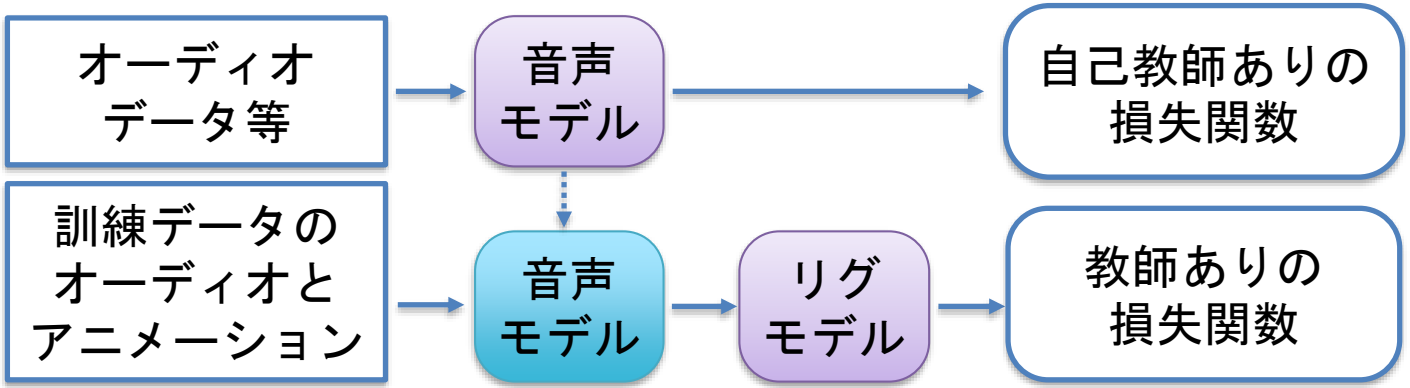


損失関数計算とスケーリング

- 訓練データと生成したデータのアニメーション トランスフォームの L1誤差
- Lipmapブレンディングを通した誤差逆伝播
- 各種の損失関数スケーリング
  - 訓練データにおける数値範囲での出力正規化
  - 訓練データがバインドポーズに近い場合は、誤差の重みを増加

音声モデルの事前学習

- 自己教師ありの音声モデルの事前学習
- オーディオデータと言語スタイルのみが必要
- 数千時間のオープンドメインオーディオを使える
- リグモデル学習でファインチューニング



自己教師ありと転移学習

- 複数の自己教師あり損失関数を試行
  - マスキングされた入力の contrastive loss
  - マスキングされたスペクトログラム予測
  - マスキングされた入力のMFCC予測
  - 高分解能スペクトログラムの主要MFCC予測
- エンド・ツー・エンド学習を超える成果は現状得られなかった

機械学習によるリップシンクアニメーション自動生成技術と FINAL FANTASY VII REMAKEのアセットを訓練データとした実装実例

Lip-Sync ML の運用事例

原 龍　　岩澤 晃

プロジェクト目標

- FINAL FANTASY VII REMAKE のアセットを使用し、次世代に向けたより良い品質とパイプラインを検討する
- 特にボイス収録時に追加されるアドリブボイスについて品質が落ちる問題について解決したい

FINAL FANTASY VII REMAKE の構成

- HappySadFace を使用したリップシンク
- ボイス数は 2.4万×4言語
  - 日本語/英語/ドイツ語/フランス語
  - カットシーンボイスは含まない数（カットシーンは手作業）
- コンバート用テキストはインゲームテキストを使用
  - バトルボイスやアドリブボイスはテキストに記載が無い

FINAL FANTASY VII REMAKE の構成

- HappySadFace を使用したリップシンク
- ボイス数は 2.4万×4言語
  - 日本語/英語/ドイツ語/フランス語
  - カットシーンボイスは含まない数（カットシーンは手作業）
- コンバート用テキストはインゲームテキストを使用
  - **バトルボイスやアドリブボイスはテキストに記載が無い**

最大の問題点

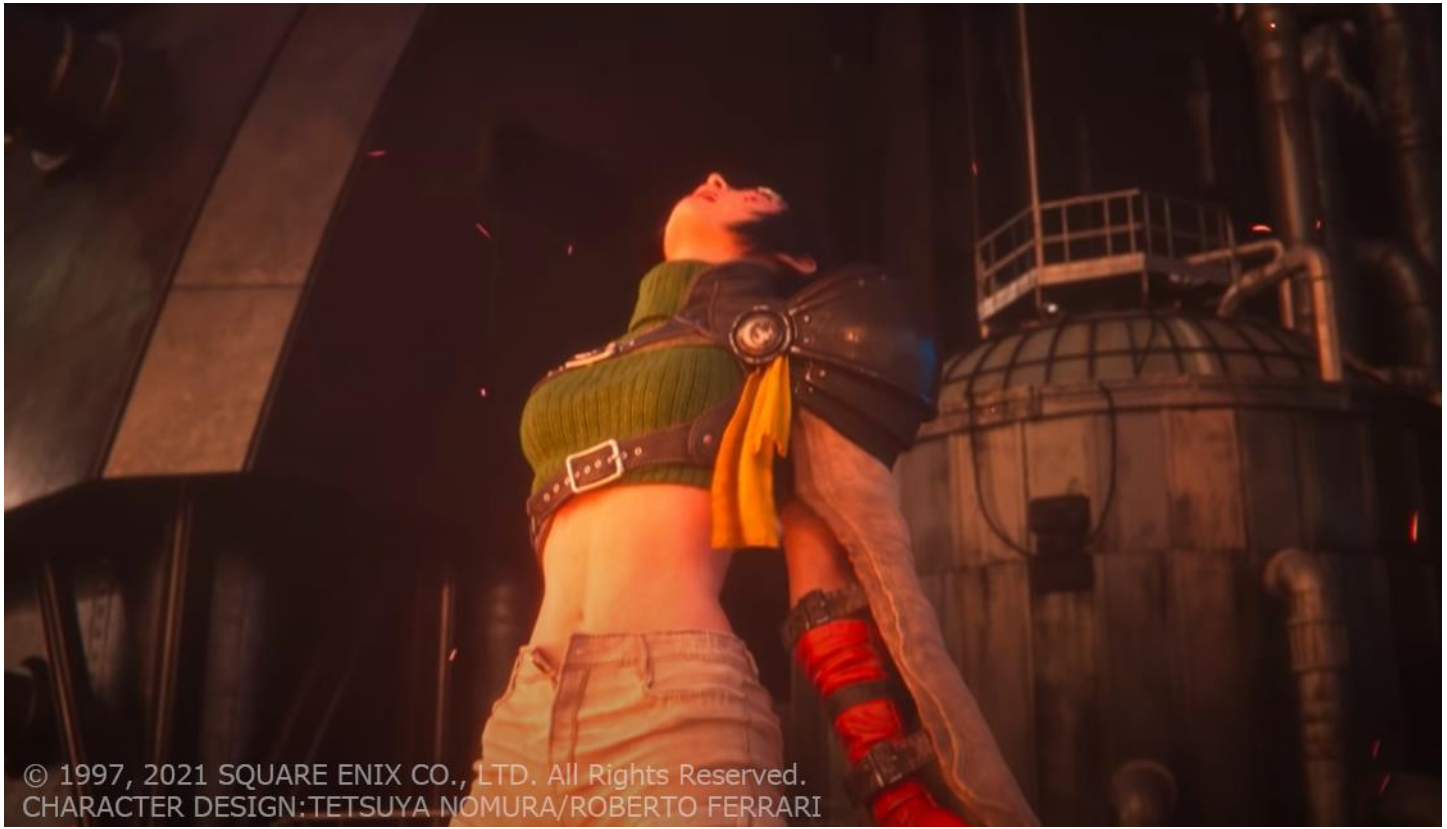
テキストが用意されない事による問題点

- HappySadFace はテキストから使用する音素を列挙し、ボイスに合わせて並べるという仕様なので、テキストが違えば実際の音素と異なるポーズウェイトが生成される
  - 笑い声や溜息などは特に相性が悪い
- 正しいテキストがあっても漢字の読みが違う事がある
  - 七番街　〇：ななばんがい　×：ななばんまち
  - 増えていくコンバート元テキストの平仮名変換辞書

テキストが用意されない事による問題点

- テキストが違う/無い場合はどうする？
  - コンバート用のテキストを差し替えるしかない
  - **押し寄せる大量のアドリブボイス向けの手動対応**
  - **ドイツ語/フランス語については語学力の問題で差し替えるテキストを用意できない**
  - テキストの差し替えを諦め、生成されたポーズウェイトを直接修正するパターンもあり

あああああああああああああああ



テキストの用意がどうやっても厳しい

- FINAL FANTASY VII REMAKE のマスターアップ後にテクノロジー推進部に課題として共有
- HappySadFace の方式ではテキストから逃れることは技術的に難しいので、方針を変更して Lip-Sync ML の開発を進めてもらう事になった

改めて HappySadFace と Lip-Sync ML の比較

項目	Lip-Sync ML	HSF
使用技術	機械学習	音素解析
入力	音声	音声、セリフテキスト
事前準備	訓練データの収集	セリフテキストの準備
クオリティ	高	低
呼吸音などの対応	○	×
対応言語	日本語、英語、 ドイツ語、フランス語	日本語、英語、 ドイツ語、フランス語
編集難易度	難	易

改めて HappySadFace と Lip-Sync ML の比較

項目	Lip-Sync ML	HSF
使用技術	機械学習	音素解析
入力	音声	音声、セリフテキスト
事前準備	訓練データの収集	**セリフテキストの準備**
クオリティ	高	低
呼吸音などの対応	○	×
対応言語	日本語、英語、 ドイツ語、フランス語	日本語、英語、 ドイツ語、フランス語
編集難易度	難	易

どうやっても厳しい…

改めて HappySadFace と Lip-Sync ML の比較

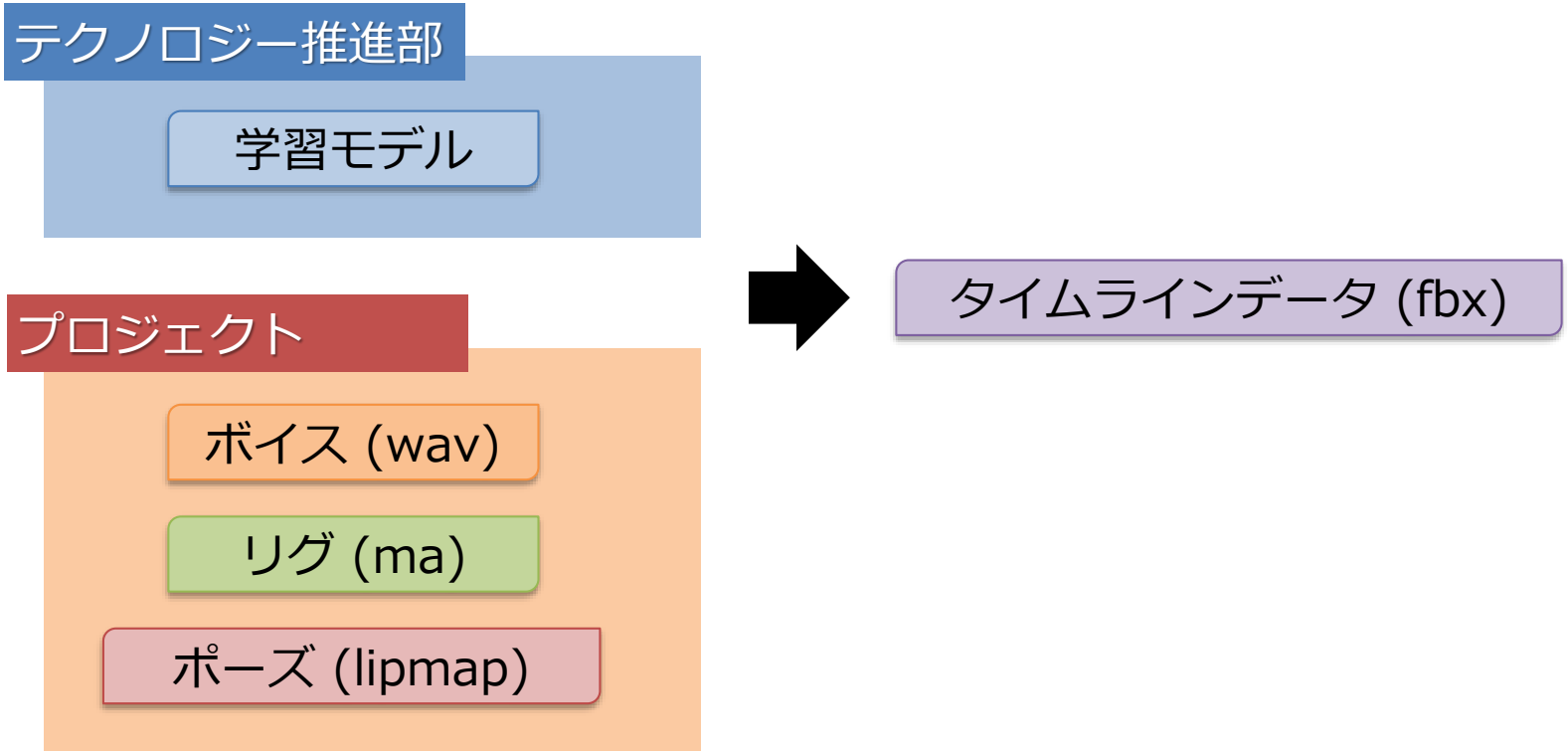
項目	Lip-Sync ML	HSF
使用技術	機械学習	音素解析
入力	音声	音声、セリフテキスト
事前準備	**訓練データの収集**	セリフテキストの準備
クオリティ	高	低
呼吸音などの対応	○	×
対応言語	日本語、英語、 ドイツ語、フランス語	日本語、英語、 ドイツ語、フランス語
編集難易度	難	易

カットシーンの手作業データが大量にある！

Lip-Sync ML 導入に向けて

- コンバートにはボイスとリグが必要
  - リグはキャラクターの Maya シーンファイルをそのまま使用
- 学習モデルはテクノロジー推進部で用意してもらう
  - 学習モデル調整は専門の機械学習エンジニアに任せた方が良い
  - 学習モデルに無いキャラクターを追加したい場合などで
    プリセットを追加する場合はプロジェクト側でも可能

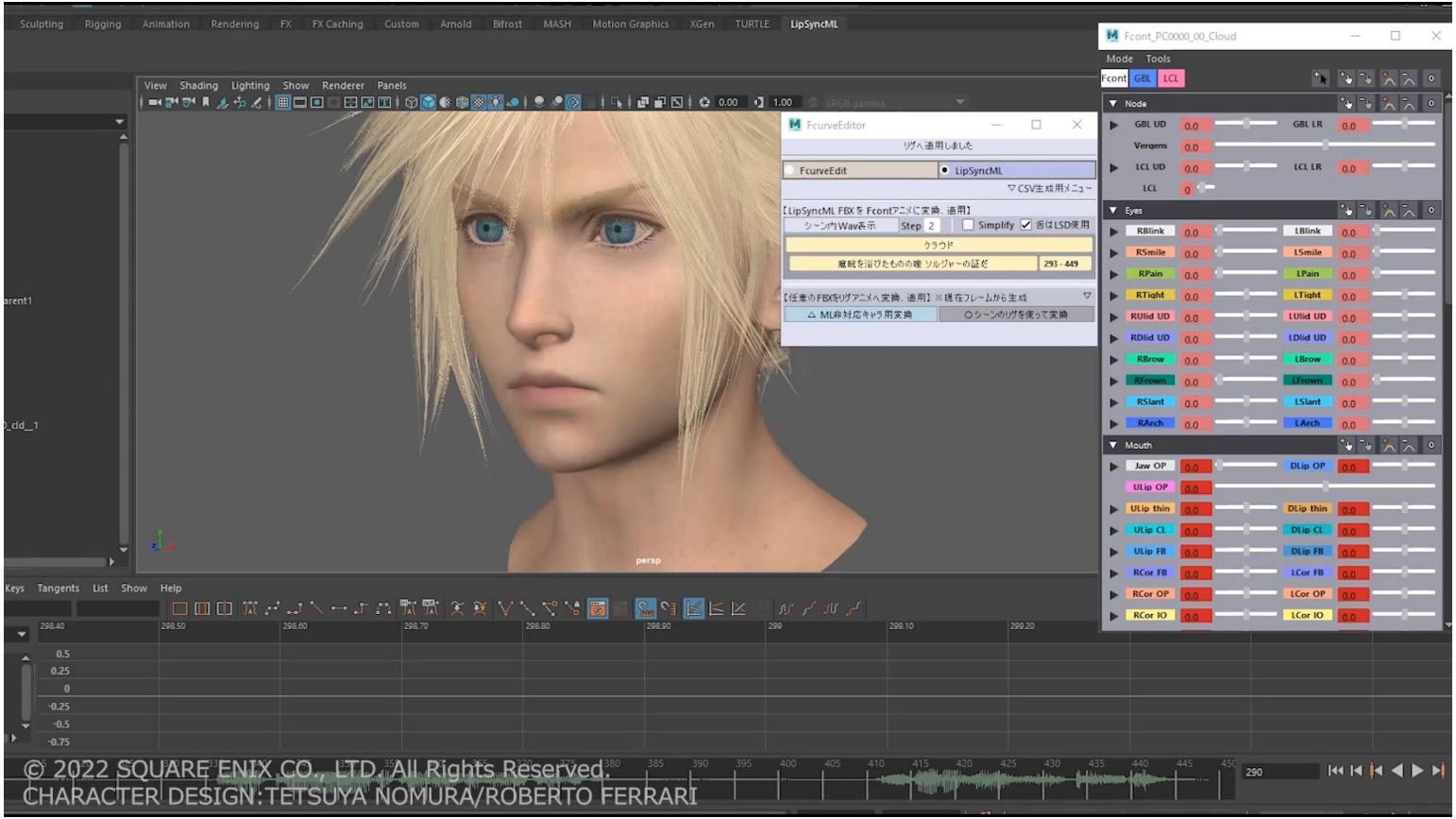
Lip-Sync ML 導入後の構成



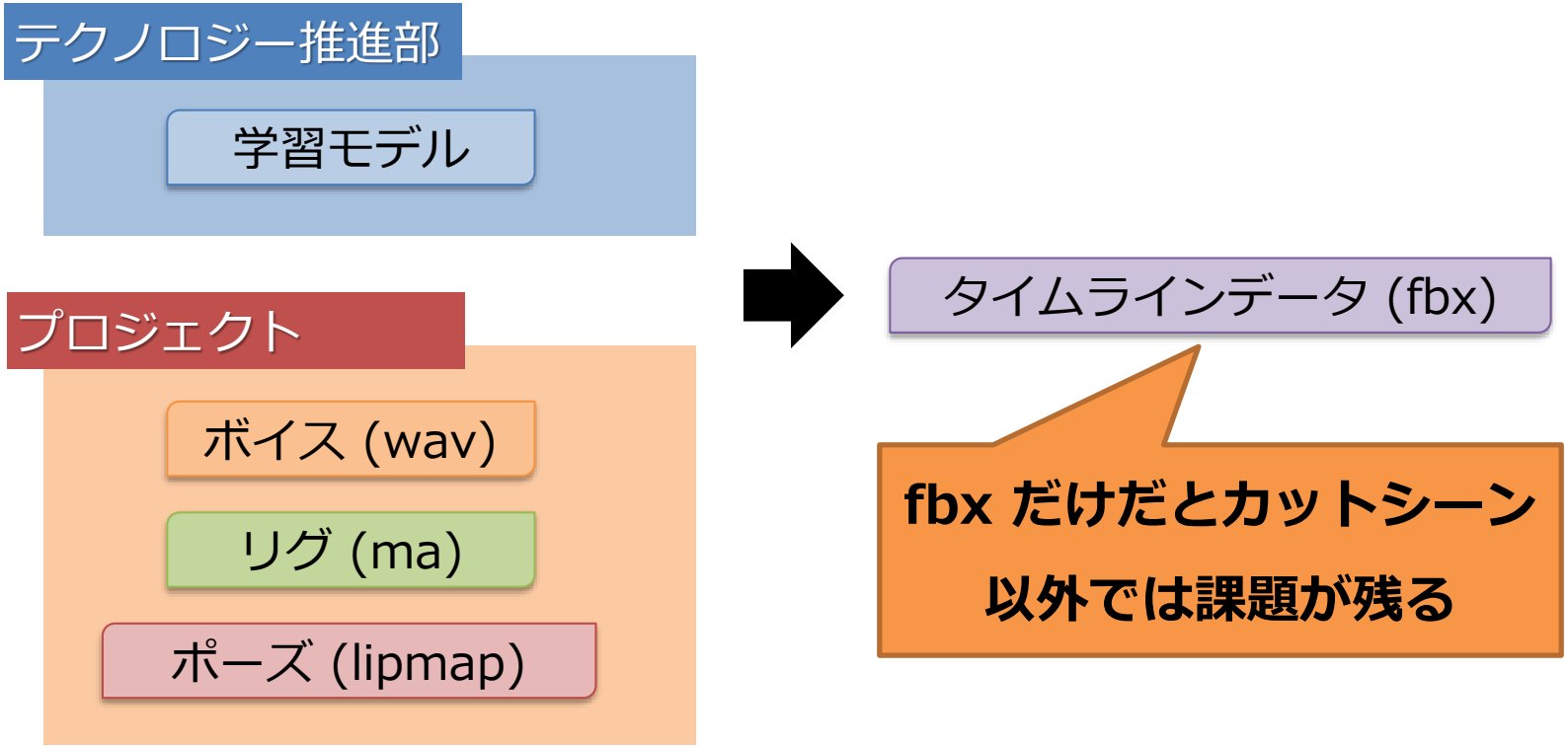
Lip-Sync ML 導入後の変化

- カットシーンのフェイシャルは手作業によるものだが、Lip-Sync ML による高品質のリップアニメが生成されるようになったため、Maya で fbx からコントローラーに戻すことで作業用データとして使用できるようになった

Lip-Sync ML の fbx からコントローラーに戻すデモ



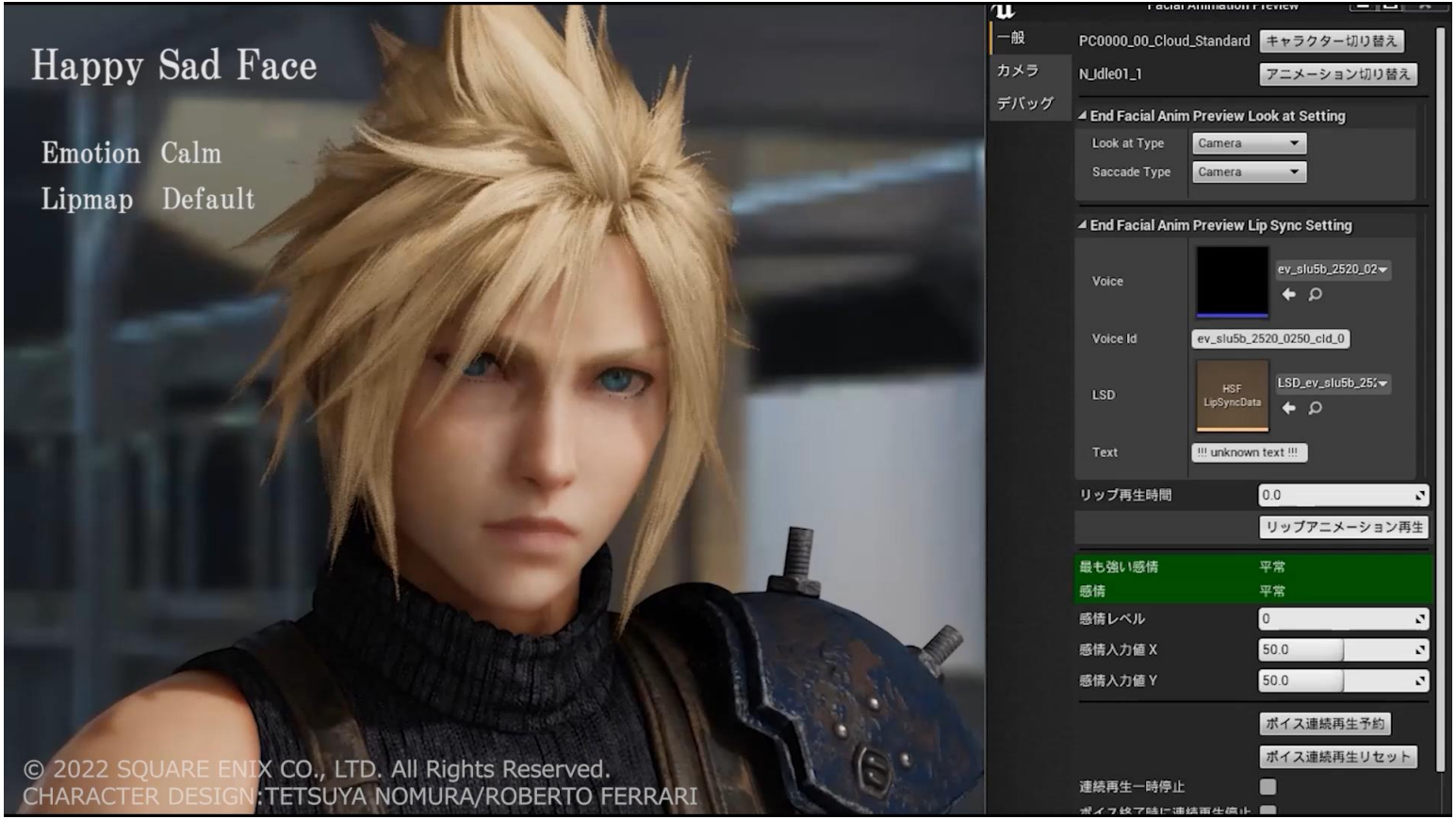
Lip-Sync ML 導入後の構成



Lip-Sync ML 導入に向けた課題

- HappySadFace の利点として、タイムラインデータがベイクされた形状データではなくポーズウェイトなので、ポーズアセットを差し替えれば形状を変えることができるというものがあった
  - これを活用することで音声から感情解析した結果を形状に反映でき、表情に合わせて出し分けることができていた

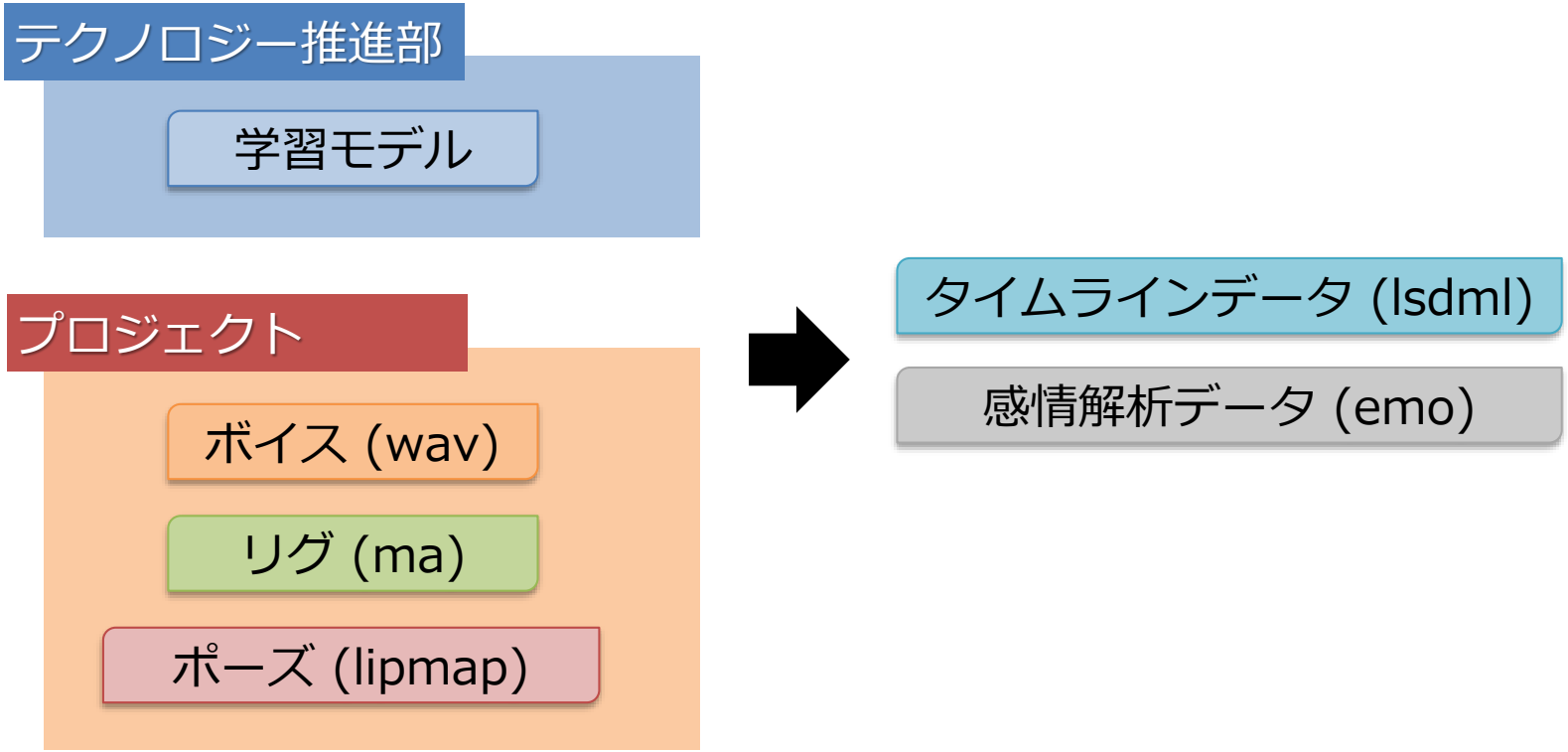
HappySadFace による感情別 Lipmap 差し替えデモ



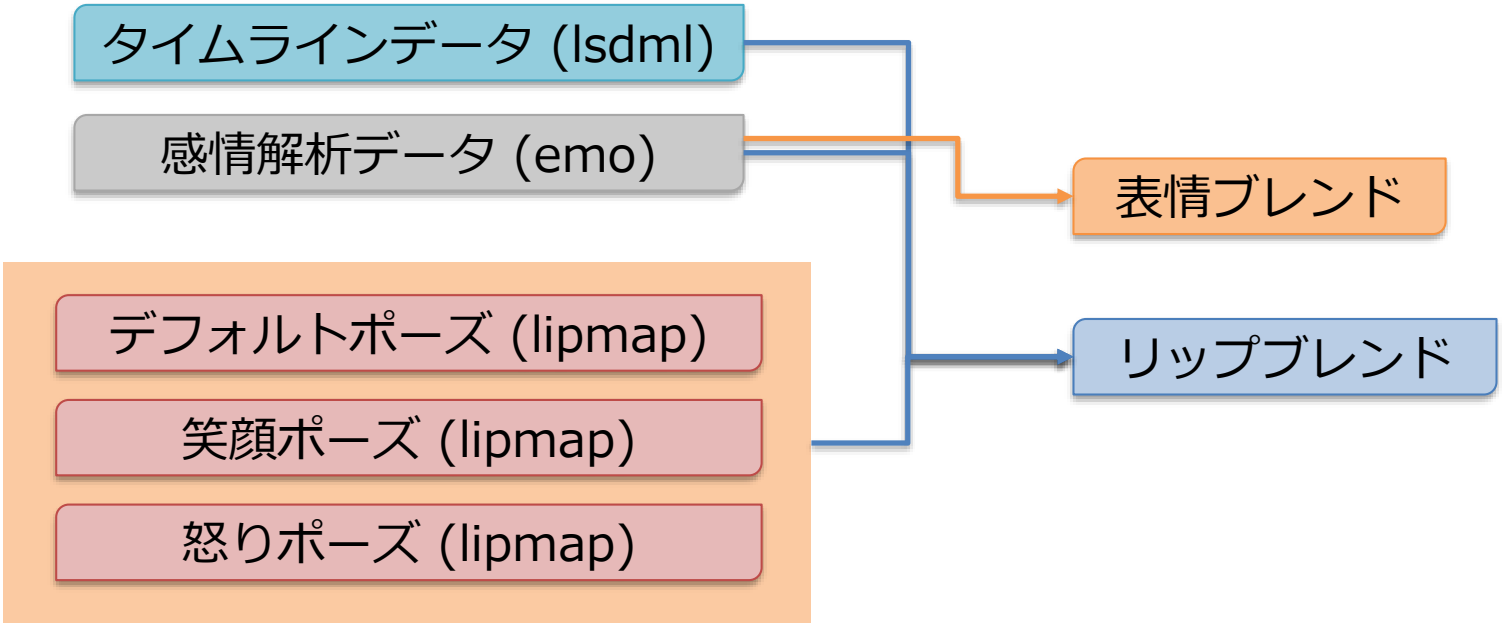
Lsdml の開発

- タイムラインデータはベイクされた形状データではなく、HappySadFace のようなポーズウェイトが理想
- Lip-Sync ML で出力された fbx を元に、ポーズウェイトとして生成した Lsdml というタイムラインデータを生成できるようにしてもらった
  - Lsdml はポーズウェイトなので品質としては最大ではないが、それよりも感情による出し分けやイテレーション効率を優先

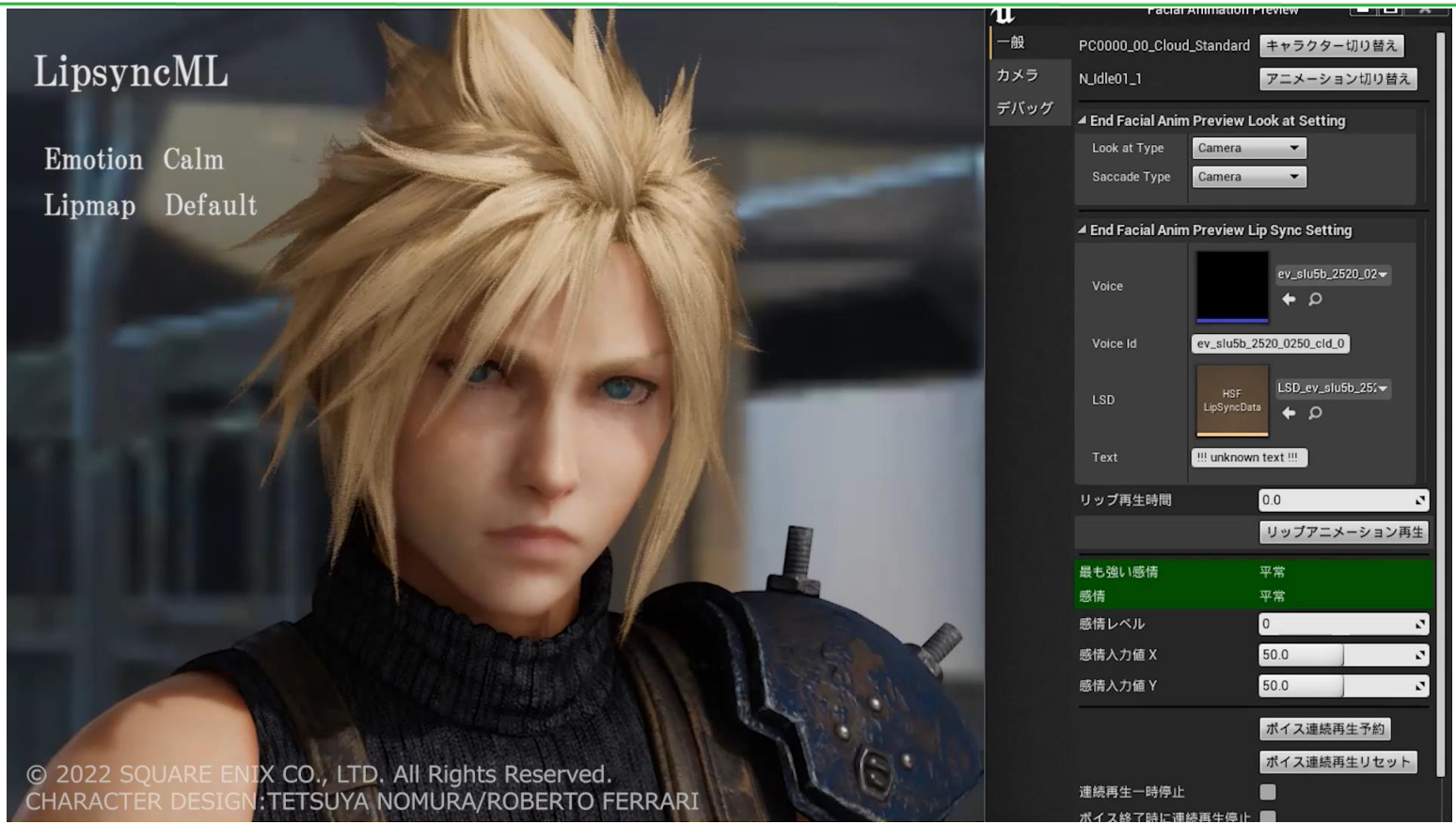
Lip-Sync ML + Lsdml 導入後の構成



Lip-Sync ML + Lsdml 導入後の構成



Lip-Sync ML + Lsdml による感情別 Lipmap 差し替えデモ



HappySadFace と Lip-Sync ML + Lsdml の比較デモ

Lip-Sync ML + Lsdml 導入後の結果

- HappySadFace のみを使用する際の課題は全て解決
  - テキスト調整地獄は去り、平穏が訪れた
- ただし新たな課題として以下のものが挙がる
  - アニメーターによる手動調整がしにくい
  - コンバート時間の増加

Lip-Sync ML + Lsdml 導入後の結果

- HappySadFace のみを使用する際の課題は全て解決
  - テキスト調整地獄は去り、平穏が訪れた
- ただし新たな課題として以下のものが挙がる
  - **アニメーターによる手動調整がしにくい**
  - **コンバート時間の増加**

アセットパイプラインで解決する

改めて HappySadFace に目を向ける

- HappySadFace は音素解析なので、いざ手動調整したい場合にタイムラインデータである LSD をアニメーターが Maya で直感的に調整しやすい
  - Lip-Sync ML だと品質は学習モデル依存かつ、出力結果に対して後から調整を入れるというのがデータフォーマット的に厳しい

つまり手動調整する場合は HappySadFace を使いたい

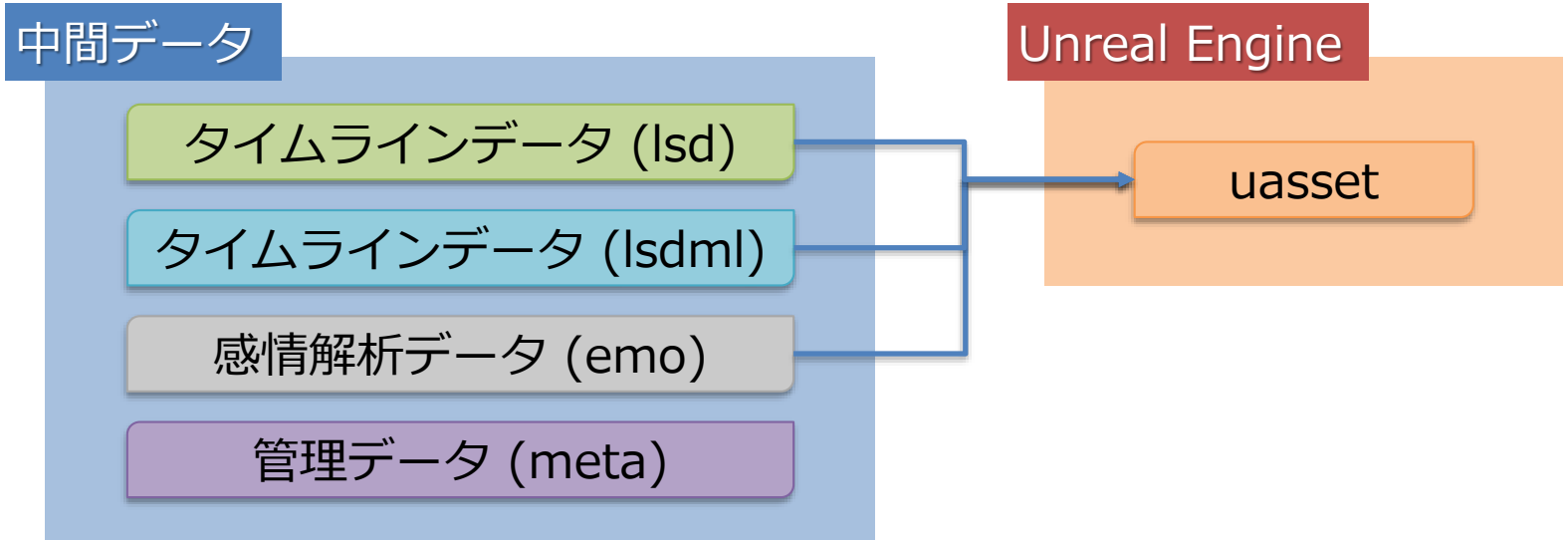
HappySadFace + Lip-Sync ML + Lsdml のアセット構成

- 中間データとしてはそれぞれ別で用意する
  - lsd (HappySadFace タイムラインデータ)
  - lsdml (Lsdml タイムラインデータ)
  - emo (感情解析データ)
  - meta (自動更新フラグやハッシュ値を保存する管理データ)
- 複数の中間データから一つの uasset としてインポート
  - Asset Import Data にそれぞれの中間データを追加することで個別にハッシュ値比較が可能になる

HappySadFace + Lip-Sync ML + Lsdml のアセット構成

- インポート済み uasset に HappySadFace と Lsdml のどちらを使うかを決定するフラグを追加
  - デフォルトは品質が高い Lsdml を使用
  - アーティストが手動調整したい場合に HappySadFace を選べる
  - 使用しなかったタイムラインデータは Cook 時に破棄して節約

HappySadFace + Lip-Sync ML + Lsdml のアセット構成



中間データを複数持つことによる利点

- ゲームからは HappySadFace と Lsdml のどちらが使われるべきなのかを意識しなくていい
  - アセットの読み分けは不要
- いずれかの中間データが更新された場合、Asset Import Data としては中間データ単位で保持しているため、ハッシュ値チェックで中間データが更新された事を検出できる
  - 定期的に再インポートジョブを実行できるので最新を保てる

コンバート時間を最適化する

- ハッシュ値チェックが有効
  - meta ファイルにコンバート元ボイスやテキストのハッシュ値を保存し、更新があるものだけをバッチに投入
  - meta ファイルのハッシュ値は HappySadFace と Lip-Sync ML では別のプロパティとして扱い、ハッシュ値の末尾にバージョン情報を追加することで学習モデル変更時の一括再コンバートなどにも対応できる
  - 一括再コンバートだと一週間以上かかるが、更新分だけコンバートされるなら数時間置きに最新の状態が保たれるようになった

meta ファイルのフォーマット

```
{
   "audioHash": "556f5ba00c9325e4ab9028d22cc18568c7ef88faeb988fb773a6c863af49d3552",
   "audioHashEmotion": "556f5ba00c9325e4ab9028d22cc18568c7ef88faeb988fb773a6c863af49d3552",
   "audioHashML": "556f5ba00c9325e4ab9028d22cc18568c7ef88faeb988fb773a6c863af49d3552",
   "rigHashML": "5e55a88e8eb32bc5641f13a614761bd5f448046d8bffe309b346694c2e23027e2",
   "textHash": "d12ac63b7df7e834e3cf63e6c6dfcb5030b305801a90d2b021bf67c33276b51c2",
   "override": 0
}
```

コンバート時のハッシュ値+バージョン番号を保存

全体のまとめ

機械学習を利用したリップシンクアニメーションの
自動生成技術と生成結果を紹介

- 機械学習とMayaツールの連携方法
  → **仮想環境を別プロセスで呼ぶ、jsonでデータのやり取りをする**
- 音声やアニメーションに関する機械学習技術
  → **スペクトログラム処理やスタイル正規化、2つの学習モデル**
- プロジェクトに導入するために必要だった実装
  → **両システムの利点を生かせる中間データでの管理**

謝辞

- 今村 紀之
- 岩渕 栄太郎
- Pijpers Jan
- Siddiq Sadjad

その他、本講演のためにご協力いただいた
すべての方々へ感謝を申し上げます

ご清聴ありがとうございました

中田 聖人
nakmasat@square-enix.com

Gracia Gil Leandro
graclean@square-enix.com

原 龍
hararyo@square-enix.com

岩澤 晃
aiwasawa@square-enix.com

SQUARE ENIX