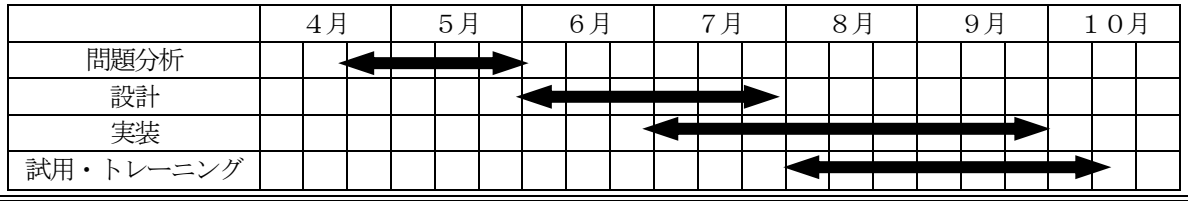
部　　門	競 技 部 門	No.1 登録番号	30002

No.2

1) 予定開発期間：５か月
2) 予定開発人数：８人



No.3

実現方法

1) 音声の解析アルゴリズム

音声の波形を一定のアルゴリズムで合成したものを復元する。合成前の音声と合成アルゴリズムは事前情報によってとても限られているので、特化した復元方法を考える。

どの読みデータをどこに配置するかについて、適切な評価関数を用いて最適なものを探す。評価関数の計算結果をもとに合成に使用された読みデータとその位置を推測する。複数の読みデータに対して評価関数を計算することになるが、これは並列化する。

評価関数の例として問題データと読みデータから同じ長さの部分を取り出し、それぞれベクトルとしてみたときの内積を考える。1 つ難点があり、数秒の問題データ、44 個程度の読みデータを想定して愚直に計算することにすると、制限時間内に計算を終わらせられないように思われる。ここで、複数の配置場所をまとめて計算することを考えると、これはベクトルの畳み込みの計算であり、高速フーリエ変換を用いて準線形時間で計算できる。これなら想定した条件において 1 台の一般的なノート PC で数秒～数十秒ですべての評価関数（内積）を計算できることを確認した。

この方法で復元が全くうまくいかなかった場合は、残りの時間で局所改善を試みる。つまり、推測された読みデータの配置をランダムに破棄し、もう一度評価関数を計算してほかの配置を試す。

想定される不都合として、評価関数を変えた結果として計算に時間がかかりすぎることが考えられる。この場合用いる情報を削る必要があるが、波形データの端を切り落とすよりはサンプルを間引くほうがよさそうである。なぜなら、綺麗な声は連続的な波形を持っていると考えられ、多少を間引いても失われる情報が少ないと考えられるからだ。

整数範囲の畳み込みは多倍長整数の計算および数え上げでも用いられており、高速なライブラリが多数存在するため、利用を検討している。

2) ユーザインターフェース

ユーザインターフェースの実装には OpenSiv3D を活用する。OpenSiv3D の機能から、ウィンドウ内において複数の画面構成を切り替えることが容易であるから、必要な機能は容易に追加できるような開発環境を作る。

3) その他（独創的なところ）

このアルゴリズムは、人間の言語の音声であることをほぼ完全に無視したモデルに対して考えられる方針である。このため、周波数解析などの、音声の特徴を取り出す有名な方針では取り出さない特徴も解析に利用することができるだろう。一方で、これは改良の余地があるということでもあり、さらなる検討が必要だ。

No.4

開発環境
OS：Windows
エディタ：Visual Studio，Visual Studio Code
言語：C++
主要ライブラリ：OpenSiv3D