Ⅰ実験

-最終レポート-

26 班 A グループ
03-160469 池田夏輝（工学部 電気電子工学科）
03-160499 武田剣志（工学部 電気電子工学科）

1. 要旨

テキストに記載のある基本的なインターネット電話を完成させ、UDP プロトコルを用いた電話のプログラムを作成したのち、独自のオプション課題に取り組んだ。
今現在、日本社会は高齢社会に突入しており、近い将来、電話で相手の声がよく聞き取れずに不便を感じてしまう人々がますます増加していくことが予想される。そこで我々のペアは、電話で届いた相手の声が通常よりもゆっくり聞こえればこの問題を解決することができるのではと考え、「相手の声をゆっくり再生する」機能を実装した。
音声波形を 2 周期分繰り返したら、再生データに 1 周期分付け足す方法を用いて 1.5 倍にゆっくり再生する方法を用い（図 1 参照）、その 1 周期を自己相関関数で算出している。また、常に音声波形を 1.5 倍で再生していると受信側での遅延がひどくなるため、送信側がしゃべっていないときには音声データを送信しない手法をとっている。
なお、このプログラムを用いるには、コンパイル時に`-pthread`, `-lm` オプションをつける必要があり、サーバー側が`./a.out “ポート番号”`, クライアント側が`./a.out “サーバーIP アドレス” “ポート番号”`を実行することにより通信が始まる。

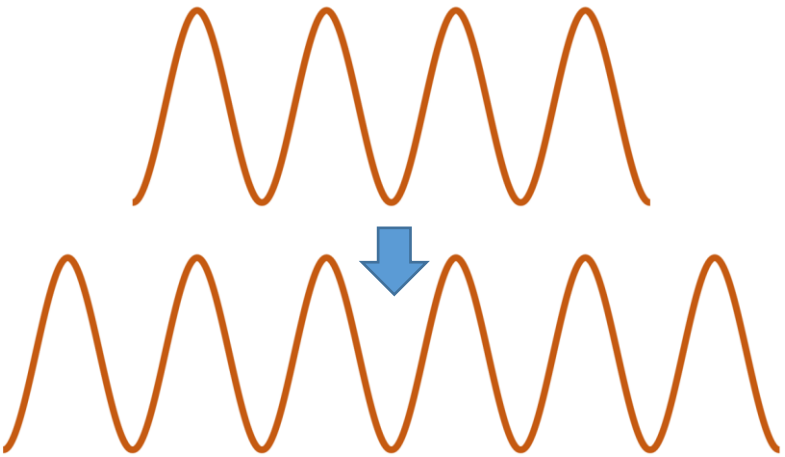

図 1

（レポートと一緒に添付しているファイルには、発表時にデモとして使用したプログラム・スライドに加えて、発表に用いたスクリプトも含まれています。）

2. 調査や実験など

2.1 プログラムの概要

まず、大まかにプログラムの流れを説明する。プログラムではコメントアウトで適宜説明している。

① 引数の数によってサーバーとクライアントを判断（サーバーならポート番号だけなので 2 つ・クライアントならポート番号に加え IP アドレスが必要なので 3 つ）し、main 関数でサーバー用処理（関数 server）、クライアント用処理（関数 client）に振り分ける。

② server 関数、client 関数ともにソケット作成（TCP プロトコルを用いている）以外には大きな違いはなく、ともにマルチスレッドで server_client_send 関数、server_client_recv 関数をたてる。
server_client_send 関数では、パソコンで録音した音を配列に入れて sound_processing_src 関数に渡し、無音判定をしたのち送信側にデータを送信する。
server_client_recv 関数では、受信した音声データを配列にいれて sound_processing_dest 関数に渡し、音の伸長と雑音除去をしたのちパソコンから音声を再生する。

2.2 sound_processing_src 関数・sound_processing_dest 関数

次に、音声処理を行っている sound_processing_src 関数、sound_processing_dest 関数について少し詳しく見ていく。

- sound_processing_dest 関数（音声データの受信後の処理）
  本来の音声データとそれを m サンプル（0.005 秒～0.02 秒分）ずらしたデータを用いて 0.01 秒分の音声データの自己相関を計算し、相関関数のピークから音声データの基本周期を求める。周期の候補は 0.005 秒～0.02 秒とした。そのあと、2 周期分繰り返したら、1 周期余分に付け加える方法を用いて 1.5 倍にゆっくり再生するための音声データを配列にしまう。
  最後に雑音除去。音声データを FFT でフーリエ変換し、周波数ごとの振幅スペクトルを求め、振幅スペクトルから一律に閾値を差し引くことによって振幅スペクトルが小さい周波数成分（つまり雑音）を除去している。
- sound_processing_src 関数（音声データの送信前の処理）
  sound_processing_dest 関数の雑音除去の処理を流用している。振幅スペクトルから引く閾値を少し大きくして、振幅スペクトルが負になる要素数を求める。この要素数が配列数の一定数以上に達したらその区間が無音区間だと判定している。

2.3 調査と実験

ペア内で取り組みたい内容を決定した後、資料探しを行い、最終的に参考文献にも挙げている、青木直史著「C 言語ではじめる音のプログラミング―サウンドエフェクトの信号処理」[1]にたどり着き、その内容を一通りさらった後、そのプログラムを参考にすることによって今回のプログラムを作成した。

2.3.1 音声データの周期の決め方

調べた結果[2]、音声信号の基本周波数が男性平均 125Hz（周期 0.008 秒）、女性平均 250Hz（周期 0.004 秒）とのことであったので、自己相関を求める際の最低周期を 0.005 秒とした。また、音声データの自己

相関を計算するのを 0.01 秒分と設定したので、その 2 倍の 0.02 秒を最高周期と定義した。

2.3.2　再生時の標本数

作成した音声処理プログラム（sound_processing_dest 関数）では、1.5 倍に変換後の音声データ配列の最後の一部が空になってしまうので、そのままだと再生する際に音がプツプツ途切れてしまう。これを防ぐために標本数を 3000 サンプル削ってから再生している。少ないサンプル数から徐々に増やしていき、耳で確認したときに一番違和感がなかったため、3000 サンプルに決定している。

2.3.3　無音区間判定や雑音除去で用いる閾値

無音区間判定や雑音除去で用いる閾値に関しても、実際に耳で聞いてみて一番違和感がなかった値を採用している。

2.3.4　malloc から配列に変更

初期のプログラムでは、プログラム内で malloc 関数を多数用いてメモリを動的に確保していたが、遅延が 4 秒以上とあまりにもひどかったため、グローバルに配列を宣言していちいち動的にメモリを確保しなくて良いように変更したところ、遅延を 1〜1.5 秒減少させることに成功した。

3. 感想・批判・実験に対する改善案等

去年 EEIC に入学してから習った内容（プログラミングや信号処理工学）で IP 電話＋α（音声伸長機能など）を実装することができ、1 年間勉強してきた成果を実感することができた。
テキストが非常にわかりやすく、順を追っていくだけで確実にソケットプログラミングまでたどり着けるように練られており、苦労することなく簡単な電話までは作り上げることができた一方、独自課題に関してはいろいろと躓くことも多く、思ったよりも実装の時間がかかってしまった。
また、ほかの人たちの発表を聞くことで、EEIC には能力がある人たちがたくさんいるということを再認識させられ、大変良い刺激となった。
実験に関して、特に不満なところはなく、今後も同じように進めていけばよいのではないかと思う。

4. 参考文献

① 青木直史
「C 言語ではじめる音のプログラミング―サウンドエフェクトの信号処理」
オーム社

② 猿渡洋
「音情報処理論 音声処理における信号処理」
http://ahclab.naist.jp/lecture/2015/sp/material/sp-v2-1.pdf

③ 「電気電子情報第一（前期）実験テキスト」
東京大学工学部 電子情報工学科 電気電子工学科