

5 画像投影を用いたデータ伝送教材の製作

森山琢海

指導教員 飯坂 寛

1. はじめに

ネットワーク上のデータ伝送の仕組みを視覚的に学ぶことは少ない。そこで、データ伝送を視覚的に提示し画像処理を使って、データ復元することで一連の流れを分かり易く学習してもらう教材を作製する。また、この製作を通してプログラミング言語の取得を目指す。

2. 研究概要

2.1 作製するシステムの概要

図 1 にシステムの流れを示す。送信側 PC から任意の ASCII 文字を 1 文字入力すると、ASCII コードに変換され、プロジェクター等にビット列が伝送されるイメージが映し出される。そのイメージは体験者（ユーザ）によってカメラで読み取られたあと受信側 PC によって画像処理され、ASCII コードは ASCII 文字に復元される。つまり、図 2 のように送信側からの文字情報がビット列で送信され、受信側でそれを画像認識と処理を行い、再び文字情報に復元するまでを視覚的に表す教材を作製した。

2.2 開発環境

開発環境は表 1 のとおりである。  
送信側の処理には JavaScript を、受信側の処理には C++を用いて開発した。

表 1 開発環境

|         |                        |
|---------|------------------------|
| OS      | Windows10              |
| 使用言語    | C++, JavaScript        |
| 使用機材    | Kinect、プロジェクター         |
| SDK     | Kinect for Windows SDK |
| オープンソース | OpenCV、Tesseract-OCR   |

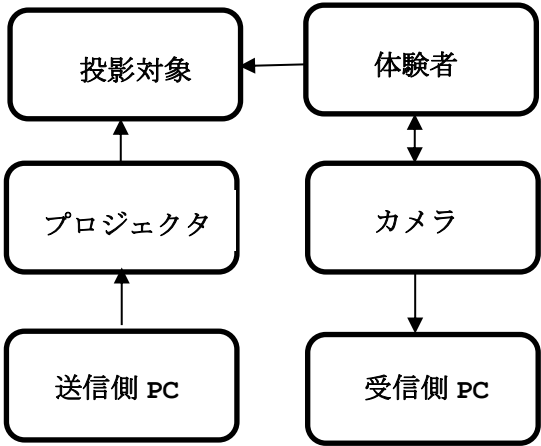


図 1 システムの流れ

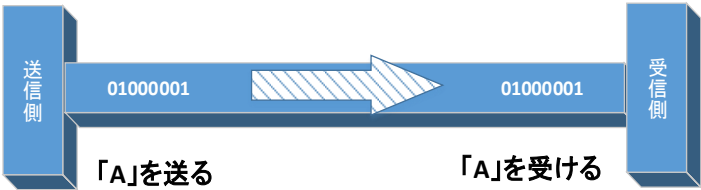


図 2 システムのイメージ

2.3 Kinect について

カメラは、マイクロソフトから発売されているジェスチャー・音声認識によって操作ができる Kinect (キネクト) を用いる。図 3 の Kinect は 2018 年現在生産を終了しており、在庫限りの販売となっている。仕様は RGB カメラ、深度センサー、マルチアレイマイクロフォン、および専用ソフトウェアを動作させるプロセッサを内蔵したセンサーがあり、プレイヤーの位置、動き、声、顔を認識することができる。



図 3 Kinect

## 2.4 Kinect for Windows SDK について

Kinect for Windows SDK は米マイクロソフトの研究部門であるマイクロソフトリサーチが開発した、言わば“マイクロソフト公式”の Windows 向け Kinect 用 SDK である。この SDK を用いれば USB 経由でパソコンに接続し、パソコンから制御可能になる。そして、Kinect センサーを活用する様々なアプリケーションを開発できるようになる。

## 2.5 OpenCV について

OpenCV とは、Intel によって開発された、画像認識に関連する機能のライブラリのことである。OpenCV は、コンピュータビジョンと呼ばれる、画像認識・解析に関するコンピューティング技術が主な用途として想定されている。OpenCV を用いることによって、例えば物体の認識、パターン認識、動作の認識といった、コンピュータビジョンに関する高度な画像処理機能を容易に利用できるようになる。

## 2.6 Tesseract-OCR について

Tesseract-OCR はヒューレット・パッカーで開発され、現在は Google で公開されているオープンソースの OCR (光学式文字認識) エンジンである。OpenCV では、OpenCV3.0 から利用可能となっている。

## 3. 送信側の仕組み

伝送される文字は簡単化のため、アルファベット 1 文字を ASCII コードに変換している。図 4 はその文字を受け付けている様子で HTML と JavaScript を用いて作成した。入力された文字は Char 型で格納され、それを整数型に変換する。その整数値を 2 進数に基数変換し、図 5 のようにブラウザ上で値を LSB から順に左から右にスクロールさせている。

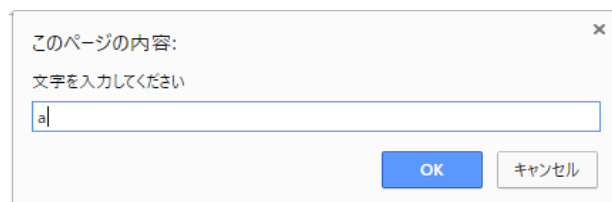


図 4 文字の入力画面



図 5 入力文字 a を ASCII コードに変換した様子

## 4. 受信側の仕組み

### 4.1 画像から文字の読み取り

図 5 のように ASCII コードに変換された画像データは、図 6 に示す受信側の流れによってビット列に復元される。モニタによって ASCII コードのビット列が表示されている画像は、カメラによって読み取られる。その読み取られた画像は、frame.tiff として保存され Tesseract-OCR で文字認識された結果、result.txt で出力される。result.txt にはビット列データが格納されている。ここでは、a の文字を送受信しているが他の文字でも同様に可能である。またモニタ出力の代わりにプロジェクター出力でも同じ処理ができる。

### 4.2 Tesseract-OCR の自動化

Tesseract-OCR をプログラムから呼び出すためにバッチファイルを使用した。バッチファイルは、Windows でのコマンドプロンプトに行わせたい命令列をテキストファイルに記述したものである。このバッチファイルは Kinect が画像を取得したタイミングで呼び出され、図 7 に記述した処理が行われる。このバッチファイルを使用し手入力が必要だったファイル処理を自動化することができた。

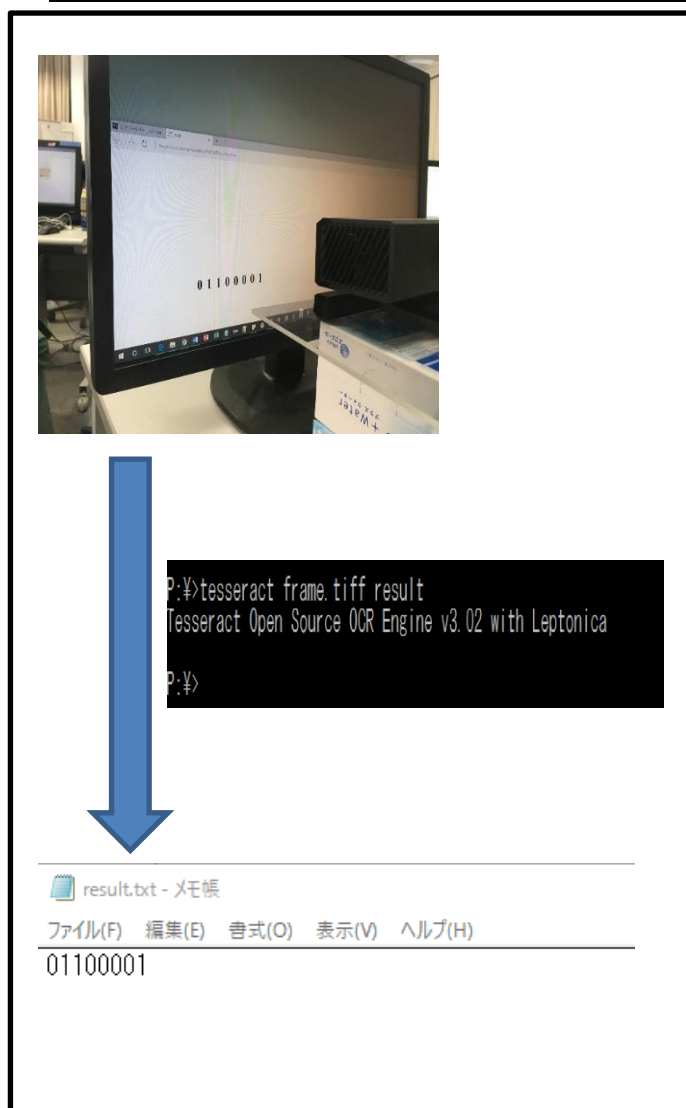


図 6 受信側の流れ

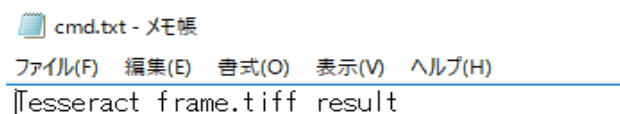


図 7 バッチファイルの内容

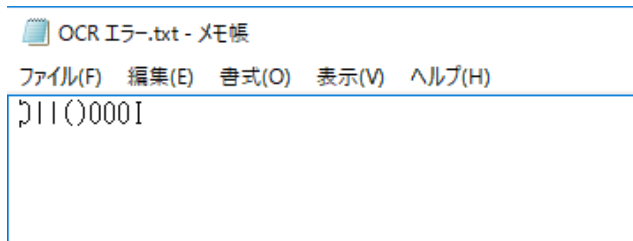


図 8 Tesseract-OCR の読み取りミス

図 8 は Tesseract-OCR が読み取りミスをした時、出力したテキストファイル内の文字列である。Tesseract-OCR の読み取り精度はカメラで撮った画像に依存しているため文字の残像や影による読み取りミスが発生する。その場合「読み取り失敗」の表示を行い、エラー処理をしている。

#### 4.3 文字の復元

送信側の仕組みで説明した手順と逆の手順で文字情報を復元させる。図 9 では、10 進数に基数変換することでフォーマット指定子から ASCII コード表に対応する文字を出力することができている。

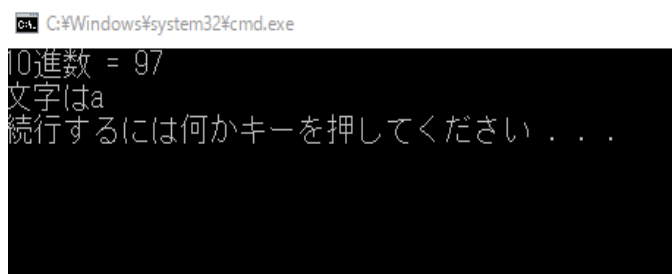


図 9 文字の復元

#### 5. 考察

文字情報を本来のデータ伝送の形に近づけ、またそれを復元して文字情報に戻すことができた。中間発表の時点で課題であった文字の復元、Kinect での画像保存機能を完成することができデータ伝送の教材としての基本的なところは完成させることができたものの、伝送中に障害が発生した場合に他のルートでデータを伝送する処理や受信側で文字列を自動的に取り込むといった機能を実装することができなかった。本来の通信における再送制御や確認応答といった整合性を持ったことが出来れば良かったと思う。投影の仕方にも体験者にもっと分かりやすく伝えるための工夫が必要だった。

#### 6. 終わりに

研究を通してデータ伝送を視覚的に分かり易くするための教材製作をすることができた。C++の基本的な書き方を学ぶことができたし、OpenCV

や Tesseract-OCR など知らなかったものもありとても新鮮だった。この研究で学んだことは今後も活かしていきたい。

## 7. 参考文献

原田建治、鈴木施哉、辻隆史、松崎俊樹、酒井大輔：プロジェクションマッピングを利用した教材(応用物理教育、2016)

Kinect for Windows SDK

<https://developer.microsoft.com/en-us/windows/kinect/develop>

中村薫、杉浦司、高田智広、上田智亜章：Kinect for Windows SDK プログラミング(秀和システム、2015)

WINGS プロジェクト：アプリを作ろう！ Visual C++入門（日経 BP 社、2016）

opencv で文字認識その 1 Tesseract ラッパ

<http://whoopsidaisies.hatenablog.com/entry/2014/11/12/003100>