

Web ブラウザを用いた遠隔操作に関する技術的検討

200912005 石原 俊

C#は文献やライブラリが豊富であり、開発効率に優れた言語である。本研究ではC#を用いてロボットの持つセンサーとモーターの制御、画像処理、コメントローラーとの通信などを実装することを目的とする。

コメントローラーに Galaxy Nexus を使い、タッチパネル上で動作を確認した。ロボットとコメントローラーは無線 LAN で接続する。汎用的に利用できるように重視し、Web ブラウザで動作させることとした。ロボットからコメントローラーへ映像を送信し、コメントローラーからロボットへモーターの制御情報を送信する。

今回は、ロボットに搭載するカメラからの画像取得を実装した。また、モーターの制御を可能とした。画像処理を非同期化し、物体検出時の画像表示を 4 フレーム毎秒から 22 フレーム毎秒に改善した。ロボット・コメントローラー間の通信には WebSocket (RFC 6455) を利用し、双方向の通信を行えるようにした。遅延の計測を行ったところ、おおよそ 150 ミリ秒程度であった。コメントローラーへの動画配信には、選択肢として HTTP Live Streaming や MPEG-DASH などがあったが、WebSocket を利用し画像を連続して送信することで、画像処理結果を合成し、かつ低遅延で表示することができた。

Haskell を用いた尤度計算プログラムの構築

200912042 中川 翔太

本研究の目的は、C++で構築した Baum-Welch アルゴリズムのプログラムを Haskell で簡潔に再構築することである。Haskell は関数型言語であり、手続き型言語である C++ とでコードの比較も行った。

尤度とは、ある前提条件（モデルのパラメータである遷移確率やシンボル出力確率）に従って結果が出現する場合に、逆に観察結果から前提条件がどうかを推定する際の尤もさである。繰り返し計算により尤度が最大になる遷移確率及びシンボル出力確率を解析的に求めるのが Baum-Welch アルゴリズムである。

C++を用いたプログラムは、多次元配列を扱う場合、多くのループ処理で階層が深くなる。また、未定義部分の計算を回避する関数や、多次元の値を 1 次元にまとめる関数などアルゴリズム以外の処理を別途書き加える必要がある。

一方 Haskell では、配列のすべての値に式を適用するリスト内包表記や、値を必要になるまで評価しない遅延評価、複数の値をまとめて扱えるタプル等の機能がある。これらを組み合わせることで、多次元配列を 1 つの式で書くことや、式を定義通り簡潔に記述することができた。

Google Maps と連携した Web アプリケーションの作成

200912043 二宮 僚洋

本研究では、地図上から今と昔の画像を比較して慣れ親しんだ土地から新たな発見をしようとする。Google Maps と連携した Web アプリケーションを作成する。

TOP ページでは、画像、位置情報、年代をデータベースから呼び出し、スライダーの指定した値に合う年代の画像を Google Maps 上に表示する。

Rails の scaffold という自動生成コマンドで投稿、編集、削除に必要なモデル、ビュー、コントローラーとデータベースが生成され、投稿、管理ページの原型ができる。アプリケーションに組み込んだ Ruby の gem ライブラリを以下に示す。

(1)paperclip で画像投稿を可能にする。(2)exifrr で画像内情報を取得する機能、Google Maps を用いて住所やクリックした場所から位置情報を取得する機能も実装し、投稿ページの投稿手順を簡素化した。(3)meta_search で検索用のページに検索機能とソート機能を実装する。(4)omniauth でログイン機能を実装し、投稿するときにログインしているアカウントを投稿者として入れるようにし、管理ページはログインしているアカウントのデータしか表示しないようにする。

結果、Google Maps 上に画像を表示する TOP ページ、画像を投稿する投稿ページ、編集や削除を行う管理ページ、検索やソートを行う検索ページ、ツイッターから認証を受けるログインページを実装し、システムを完成させた。

Tesseract-ocr を用いた車両ナンバープレートの検出

200912027 師社 佑也

工場で勤務している社員が出社、退社する際にタイムカードを押すという動作がある。本研究の目的は、車両ナンバープレートから文字情報を読み取ること、Text データとして管理するで、タイムカードを押すという手間を減らすことである。実装には OpenCV の .NET ラッパーである Emgu CV と C#を用いる。

Tesseract-ocr は標準で日本語データを 4100 文字ほど用意している。しかし、ナンバープレートの認識において十分な結果を得られなかった。そこで、3 枚の画像から 201 文字の画像データを作成した。3 枚の画像には、ナンバープレートで用いられる漢字、平仮名、数字がそれぞれ記入されている。Tesseract-ocr でトレーニングすることで文字認識させるための学習データを作成した。標準の英語データの学習方法に倣い、日本語も文字列を傾けて学習を行った。また、書体はナンバープレートと同じゴシック体のメイリオを使用した。

標準の日本語データと作成した学習データを比較した。その結果、標準データでは認識された「多摩」[500]「4」[6]が作成したデータを用いた場合、正しく認識するようになった。