

Tesseract-ocr を用いた車両ナンバープレートの検出

200912027 師社佑也

本研究では、Google が無料で提供している OCR エンジン Tesseract-ocr を用いて、車両ナンバープレートを自動で読み取る。実装は OpenCV の .NET ラッパーである EmguCV と C#を用いる。

処理の流れは、初めにカメラから車両ナンバープレートの検出を行う。次にグレースケール画像に変換し、Canny アルゴリズムを用いて画像のエッジ検出を行う。最後に Tesseract-ocr を用いることで地名と車両番号をテキストデータとして取得できる。

今回は車両から外されているナンバープレートの画像を使い、ナンバープレートを検出する作業を省略し、エッジ検出とテキストデータ化について行った。

結果は、正しく検出はできなかった。問題点として、「多摩」という文字が断片的に表示されたこと、複数段で構成されている場合、どちらから読み取るのか指定しなければいけない。

今後の課題として、上下別々に文字検出を行うことで誤検出を防ぐことができれば試す必要がある。

jQuery を用いた新潟県の路線図の作成

200912007 井野 宏陽

本研究では、jQuery を使って新潟県の路線図を作成することが目的である。

jQuery は、2006 年に JavaScript ライブラリの 1 つとして発表され、簡単な記述で Web ページに効果を付け足すことができる。jQuery は指定した要素に何らかの操作が発生したタイミニングで指定した要素の HTML や CSS を変更する。

現段階ではまだ新潟県の路線図の作成に進めていないので開発前の段階として今回は、ボタンをクリックするとページ上の文字が書き変わるものと、テーブルヘッダーをクリックすると昇順、降順が切り替わるプログラムを作成した。

Google Maps と連携した Rails アプリケーションの構築

200912043 二宮 僚洋

本研究では Google Maps と連携した WEB アプリケーションを Rails で構築する。今回は Google Maps 上に年代別に画像を表示するアプリケーションおよび画像投稿フォームを作成した。

Rails はスクリプト言語 Ruby で書かれた WEB アプリケーションフレームワークである。MVC(Model・View・Controller)アーキテクチャをサポートしており、モデル、ビュー、コントローラを自動生成できる。このため、短期間で WEB アプリケーションを構築できる長所を持っている。

はスライダーで年代を設定し、該当する画像を Google Maps 上に表示する。投稿フォームは scaffold コマンド、画像管理は paperclip プラグインでテンプレートと自動生成した。テンプレートを修正し、Google Maps から位置情報を取得する投稿フォームを作成した。

今後は、見栄え、処理効率を改良する。

ウェブブラウザを用いたロボットの操縦インターフェースの作成

200912005 石原 俊

本研究ではレスキューロボットの操縦インターフェースを作成する。これは機械学科と共同プロジェクトの一部である。主にタブレット端末で利用されることを想定しており、複数のタブレットフォームで動作させるためにウェブブラウザ上で操作できるようにする。

ウェブブラウザ上で動作することで、操縦インターフェースをタブレットフォーム毎に開発する必要がなくなり、多くの端末から操作することが可能になる。ロボットにはウェブサーバーと無線 LAN アダプターが搭載されている、タブレットでアクセスポイントへ接続し、ウェブブラウザから操縦する。

カメラ映像のストリーミング配信では 8 秒程度の遅延が発生した。また、低遅延な配信は一部のタブレットフォームしか対応していない。このため、WebSocket を用いて JPEG 画像を連続して送信することで、低遅延な表示を実現した。また物体識別の 1 つとして顔認識を行ったが、顔認識は処理時間が長いので、画像の送信が大きく遅延している。

今後の課題として、顔認識処理の非同期化を行う。また、物体に応じた操作メニューの動的生成や、ロボットのステータス表示を実装する。

Haskell を用いた尤度計算プログラムの構築

200912042 中川 翔太

本研究は尤度計算法の Baum-Welch アルゴリズムを、関数型言語の Haskell で構築する。比較対象として手続き型言語の C++ でもプログラムを構築した。

Baum-Welch アルゴリズムは、繰り返し計算を行うことモデルの遷移確率とシンボル出力確率を推定できるアルゴリズムである。例えば PC カメラが得た情報など、状態が時間経過によって変化し、出力が各状態で確率的に決まるものを扱い尤度計算をする際に用いられる。将来的にレスキューロボットへ搭載し、助けを求める人が居るかどうかを判断するための利用を目的としている。

C++を用いたプログラムは、多次元配列を扱う場合など、ループ処理が必須になりコードが長く複雑になることがしばしばあった。コードの長さや複雑さは、プログラムの分りづらさや、記述ミスの見つけにくさの原因になる。

一方 Haskell には、リスト (配列) のすべての値に処理を適用する機能や、多次元配列の一部だけを抜き出す機能が用意されている。また、関数型言語の特徴として値や式はすべて関数として定義される。これらを組み合わせることで、多次元配列やループを扱うコードを非常に短く、簡潔に記述することができた。

今後は、多次元配列やループに限らず、プログラムを Haskell でそれほど簡潔に記述できるかを明らかにしていく。