2D4-2

Convolutional Neural Network による物体認識の自己位置推定への統計的活用

Statistical Localization Exploiting Object Recognition by Convolutional Neural Network

石伏 智 *1 Satoshi Ishibushi
谷口 彰 *1 Akira Taniguchi
高野 敏明 *1 Toshiaki Takano
谷口 忠大 *1 Tadahiro Taniguchi

立命館大学 *1
Ritsumeikan University

In this paper, we proposed a Monte-Carlo localization method which exploits object recognition by Convolutional neural network (CNN) for autonomous vehicle. CNN is known as one of Deep learning method. In many cases, Monte-Carlo localization method uses control data and measurement data. However, some errors often be observed in location estimation . We proposed that autonomous vehicle employs object recognition results by CNN as one of measurement data using Bag-of-Features representation. The experiment result shows that the proposed method can reduce estimation error.

1. はじめに

近年，移動ロボットの研究分野では人間や動物のように自律的に環境を動き回ることのできるロボットを実現しようとする研究が行われている．このような自律移動ロボットを開発するには，ロボットに障害物や物体などの周囲の環境情報をセンサから知覚させ，環境の地図およびロボット自身の位置を推定させることが重要となる．ロボットが地図と自己位置を同時に推定することを SLAM（Simultaneous Localization and Mapping) という [Thrun 05].

一方，画像認識の研究分野では，ロボットに画像や物体を認識させる方法として Deep learning が注目を浴びている．Deep learning は三階層以上の深い構造を持つニューラルネットワークのことである．近年，Deep learning によって得られた内部表現を用いた物体認識の手法が物体認識の分野で成果を出している [岡野 13]．Deep learning の手法の一つである CNN(Convolutional neural network) は入力画像の局所的な特徴を受け取る畳込み層と受け取った特徴の一部を出力するプーリング層が積み重ねられて構成されたニューラルネットワークである [Krizhevsky 12]，[岡谷 13].

自己位置推定の手法として最も有名な MCL(Monte-Carlo Localization) では多くの場合，機体の制御情報と外界の障害物や壁までの距離を示す観測情報を用いている．制御情報や観測情報には計測誤差が生じることが多い．例えば，車輪のスリップや地面の段差によってロボットの制御情報に実際の移動距離との誤差が生じる．これらの誤差はベイズフィルタの原理により局所的には補正することができる．しかし，大域的な補正ができない場合が多く，推定結果の事後分布が多峰的になることがある．本稿では CNN により得られた物体認識結果を用いることで，大域的な位置推定の誤りを減らすことを目指す．距離を示す観測情報に加え，CNN による観測した画像の物体認識結果を統計的に統合して用いることによって，自己位置推定を行うことを提案する．

連絡先: 石伏 智，立命館大学情報理工学研究科，滋賀県草津市野路東 1-1-1 立命館大学情報理工学部,ishibushi@em.ci.ritsumei.ac.jp

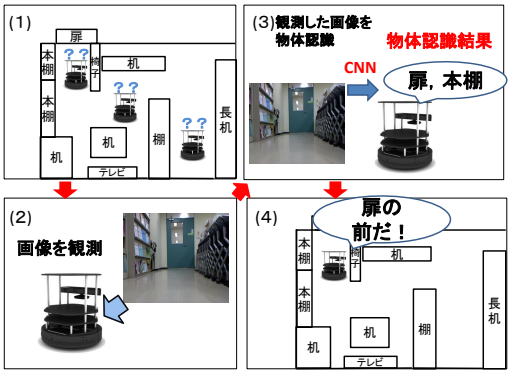

図 1: Turtlebot2*1 を用いた本研究のタスクの概略図

2. 提案手法

2.1 タスクの概要

地図のある環境上で移動ロボットを移動させ，自己位置推定を行わせる．提案手法のタスクの概要を示す図 1 を例に説明する．MCL では図 1(1) のように複数の位置を自己位置の候補としてしまうことがある．本研究では図 1(2)，(3) のように観測した画像を CNN によって物体認識した結果を観測情報の一つとする．図 1(2) の場合では周りの環境の画像を観測し，図 1(3) のように CNN によって「扉」,「本棚」などの物体認識結果を得る．このことにより，図 1(4) のようにロボットは自己位置が入口の付近に存在することを推定できる．

2.2 位置推定手法

谷口らは不確実な音声認識結果とロボットの自己位置推定情報を統合した自己位置と語彙の同時推定モデルを提案している [谷口 14]．谷口らのモデルを参考にし，MCL に場所領域のインデックス C_t と特徴ベクトル f_t を付与し拡張する．C_t と f_t を加えたグラフィカルモデルを図 2 に示す．グラフィカルモデルの各要素についてまとめたものを表 1 に示す．本研究はロボットの自己位置の近さと観測した画像の特徴の近さを

*1 http://www.turtlebot.com/

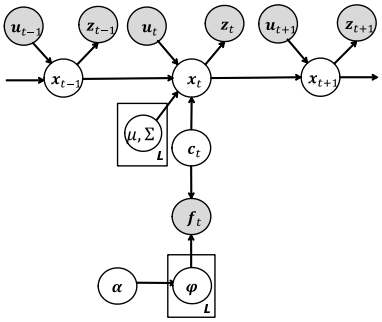

図 2: 提案手法のグラフィカルモデル

表 1: グラフィカルモデルにおける変数一覧

x_t	ロボットの姿勢
u_t	制御値
z_t	距離の計測値
f_t	観測した画像の特徴ベクトル
C_t	場所領域のインデックス
μ, Σ	ガウス分布の平均値と共分散
φ	多項分布のパラメータ
α	ディリクレ分布のハイパーパラメータ

考慮した範囲を一つの場所領域と定義する．周りの環境の画像の物体認識結果と自己位置の近さから場所領域が決められる．それぞれの場所領域のインデックスを C_t とし，以下のように定義する．

$$C_t \in \boldsymbol{C} = \{1, 2 \cdot \cdot, L\} \tag{1}$$

ここで L はあらかじめ決めた場所領域の個数である．また本研究では t 時刻で，CNN によって物体を認識した結果を場所領域で得られた特徴ベクトル f_t と定義する．物体の種類はさまざまあるため，f_t は以下のように定義する．

$$\boldsymbol{f}_t = \{f_t^1, f_t^2 \cdots f_t^I\} \tag{2}$$

ここで I はあらかじめ CNN が学習した物体のクラスの数である．

MCL の導出式に特徴ベクトル f_t 加えた式を以下に示す．

$$\begin{aligned} &p(x_{0:t}|z_{1:t}, u_{1:t}, f_{1:t}) \\ &\quad \propto p(z_t|x_t)p(f_t|x_t)p(x_t|x_{t-1}, u_t) \\ &\qquad \times p(x_{0:t-1}|z_{1:t-1}, f_{1:t-1}, u_{1:t-1}) \end{aligned} \tag{3}$$

式 (3) から導かれる $p(f_t|x_t)$ は x_t の位置で，場所領域で得られる特徴ベクトル f_t を観測する確率を意味する．f_t が得られる確率を C_t ごとに求めて周辺化する．このときの式を以下に示す．

$$p(f_t|x_t) = \sum_{C_t} p(f_t|C_t)p(C_t|x_t) \tag{4}$$

$$\propto \sum_{C_t} p(f_t|C_t, \varphi)p(x_t|C_t, \mu, \Sigma)p(C_t) \tag{5}$$

$$= \sum_{C_t} p(f_t|\varphi_{C_t})p(x_t|\mu_{C_t}, \Sigma_{C_t})p(C_t) \tag{6}$$

ここで φ_{C_t} は C_t 番目の多項分布のパラメータであり，μ_{C_t}, Σ_{C_t} は C_t 番目のガウス分布の平均と共分散である．多項分布，ガウス分布はそれぞれ場所領域の個数だけ用意する．

$p(f_t|\varphi_{C_t})$ については以下のように多項分布で求める．

$$p(f_t|\varphi_{C_t}) = \mathrm{Mult}(f_t^1, f_t^2, \cdots, f_t^I|\boldsymbol{\varphi}, K) \tag{7}$$

ここで K は観測値の個数であり，以下のような制限を付ける．

$$K = \sum_{i=1}^{I} f_t^i \tag{8}$$

また，$p(x_t|\mu_{C_t}, \Sigma_{C_t})$ については以下のように多次元ガウス分布で求める．

$$\begin{aligned} p(\boldsymbol{x_t}|\mu_{C_t}, \Sigma_{C_t}) &= \frac{1}{(2\pi)^2|\Sigma_{C_t}|^{1/2}} \\ &\times \exp\left\{-\frac{1}{2}(\boldsymbol{x} - \mu_{C_t})^T\Sigma^{-1}(\boldsymbol{x} - \mu_{C_t})\right\} \end{aligned} \tag{9}$$

ここで μ_{C_t}, Σ_{C_t} はそれぞれ平均値と共分散を示す．

$p(C_t)$ は無情報と仮定し以下のように一様な値とする．

$$p(C_t) = \frac{1}{L} \tag{10}$$

また，本研究ではロボットの姿勢 x_t は x, y 平面での位置座標とロボットの方向 θ を示すため，$(x, y, \sin\theta, \cos\theta)$ と定義する．θ は x 軸の方向を 0°，y 軸の方向を 90° として定義する．

2.3 CNN による物体認識結果の活用

本研究では物体認識として CNN を用いる [Krizhevsky 12], [岡谷 13]．CNN は畳み込み層とプーリング層と呼ばれる 2 種類の層を交互に積み重ねた構造の多層ニューラルネットワークである．CNN は上記のような構造をもつことにより，得られる特徴に不変性があることで知られている．CNN の模式図を図 3 に示す．また，CNN の最後のプーリング層の後にユニット間をすべて結合した層を配置し，最後の出力に以下のようなソフトマックス関数を用いることにより，入力画像がそれぞれのクラスの物体 O_i である確率 $p(O_i)$ を求めることができる．

$$p(O_i) = \frac{e^{x_i}}{\sum_{k=1}^{N} e^{x_k}} \tag{11}$$

ここで，$x_i (i = 1, \cdots, I)$ は最終層への入力を示しており，I は物体のクラスの数を示す．本研究では，特徴ベクトル f_t の各要素は CNN によって得られた画像を物体として識別した確率 $p(O_i)$ を用いる．この際 $p(O_i)$ を 10^s 倍した値を整数にすることでカウント数とし，Bag-of-Features 表現としてとり扱う．具体的に述べると，$p(O_i) = 0.2$ と得られた場合で $s = 2$ とすると，物体 O_i を示す特徴ベクトルの項 f_t^i が 20 回観測されたものとする．

CNN による入力画像の認識結果の例を図 4 に示す．図中に示す単語はそれぞれの物体のクラスラベルを示しており，一つのクラスラベルに複数の単語がつけられている場合もある．例えば，mailbag，postbag は同じ物体として扱い，両方の単語を合せて一つのクラスラベルとなっている．また，クラスラベルの右に示す数値は，入力画像がそれぞれのクラスの物体である確率を示している．図 4 では確率が高かった上位五つを上から順に並べており，一番確率が高かった物体のクラスは turnstile(改札口) であり，確率が 0.0691 を示している．

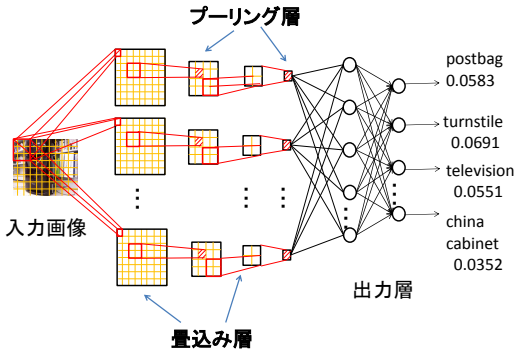

図 3: CNN の模式図

turnstile | 0.0691
mailbag, postbag | 0.0583
china cabinet, china closet | 0.0558
television, television system | 0.0551
pay-phone, pay-station | 0.0352

図 4: CNN で認識した結果の例（上）認識対象の画像 (下）出力結果のラベルと確率値

図 4 の入力画像画像では実際は空気清浄機の画像を示している．しかし，認識結果として確率が高いクラスは turnstile(改札口) や mailbag(郵便袋）といった別の物体を表すクラスとなっている．しかし，一定してこのような画像を turnstile や mailbag であると認識することにより，このような画像が観測される場所領域では turnstile，mailbag の要素が高い特徴ベクトルを得られるといった学習を行える．このことにより turnstile，mailbag という要素が高い特徴ベクトルが得られた場合，空気清浄機が配置されている付近に自己位置が存在する確率が高くなる．

また，多項分布のパラメータ φ_l はディリクレ事前分布と観測情報からディリクレ事後分布を計算して，サンプリングによって得る．

$$\boldsymbol{\varphi}_l \sim Dir(\boldsymbol{\varphi}_l|\boldsymbol{\alpha}+\boldsymbol{m}) = \frac{\Gamma(\alpha_0+K)}{\Gamma(\alpha_1+m_1)\cdots\Gamma(\alpha_I+m_I)}\prod_{i=1}^{I}\varphi_{l,i}^{\alpha_i+m_i-1} \quad (12)$$

ここで $\boldsymbol{m}$ は各特徴ベクトルが観測された回数を示しており，$\boldsymbol{m}=(m_1,\cdots,m_I)^T$ である．また，$\boldsymbol{\alpha}$ はディリクレ分布のハイパーパラメータであり $(\alpha_1,\cdots\alpha_I)^T$ を表す．

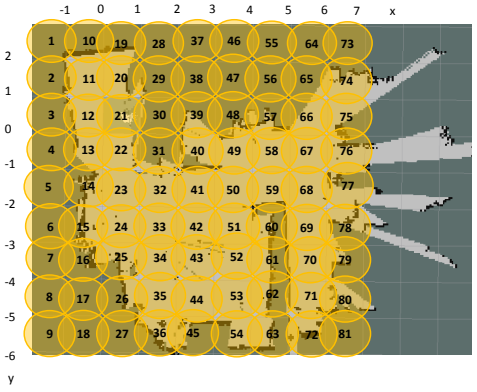

図 5: x, y に関する場所領域に対応するガウス分布を配置した地図の一例

3. 実験

3.1 実験方法

本研究の提案手法の有効性を測るために，CNN による物体認識を加えた MCL と物体認識を加えない MCL で位置推定を行った場合の，それぞれの自己位置推定結果を示すパーティクルの状態について比較した．実験の流れとしては，まず SLAM により地図を生成する．生成した地図を元に自己位置推定を行い，各場所領域の多項分布のパラメータ φ を学習する．そして，学習結果をもとに提案手法での位置推定を行う．

3.2 実験条件

3.2.1 自己位置推定について

本研究での実験はすべて実機の Turtlebot2 を用いて，場所は立命館大学創発システム研究室で行った．自己位置推定のパーティクル数は 1000 とした．Turtlebot2 を自己位置推定を行わせながら，ワイヤレスコントローラを用いて移動させて実験を行った．ワイヤレスコントローラにはソニー・エンターテイメント社の DUAL SHOCK 3 を用いた．

3.2.2 学習済み CNN ツール:Caffe

本研究では CNN のツールとして Caffe を用いる．Caffe はカリフォルニア大学バークレー校の研究センターである BVLC[*2] が中心となって開発しているオープンソースソフトウェアである．Caffe は学習済みのリファレンスモデルが配布されているので本研究ではそれを用いる．学習された物体の種類は $I=1000$ であり，椅子やボールといった物体から，象やカンガルーなどの動物まで学習している．

3.2.3 場所領域の決め方

本研究では，x, y, θ の値が近ければ，似たような特徴が観測できると仮定し，位置 x, y と方向 $\sin\theta, \cos\theta$ の近さから場所領域が決められるものとする．本研究では x, y 座標と $\sin\theta, \cos\theta$ に関する四次元ガウス分布によって場所領域を決める．グリッド状にガウス分布を配置した地図を図 5 に示す．図中の黄色の円が各ガウス分布を表現している．図では番号 1 の x, y に関するガウス分布の中心を $x_1=-1.5, y_1=2.5$ とし，番号 n の x, y に関するガウス分布の平均ベクトルの要素 x_n, y_n について以下のように決めた．

$$(x_n, y_n) = (x_1 + \left\lfloor \frac{n-1}{9} \right\rfloor, y_1 - n + 9\left\lfloor \frac{n-1}{9} \right\rfloor + 1) \quad (13)$$

*2 http://bvlc.eecs.berkeley.edu/

それぞれの x, y に関して決めた範囲には四方向に関するガウス分布が存在する．具体的には x 軸の正の方向を 0°，y 軸の正の方向を 90° として，中心を $\sin\theta = 0, \cos\theta = 1$ とするガウス分布を A，$\sin\theta = 1, \cos\theta = 0$ とするガウス分布を B，$\sin\theta = 0, \cos\theta = -1$ とするガウス分布を C，$\sin\theta = -1, \cos\theta = 0$ とするガウスを D として，四方向のガウス分布が存在する．

例えば，図 5 の番号 1 に存在するガウス分布 A を場所領域のインデックス $C_t = 1$，図 5 の番号 1 に存在するガウス分布 B を $C_t = 2$ とするような場所領域の決め方を行う．ここで上記の場所領域のインデックス $C_t = 1$ のガウス分布の平均値は $(\mu_{1,x}, \mu_{1,y}, \mu_{1,\sin\theta}, \mu_{1,\cos\theta}) = (-1.5, 1.5, 0, 1)$ となる．また各場所領域のガウス分布の共分散の値は固定値として以下のように設定した．

$$\Sigma = \begin{pmatrix} \sigma_{xx} & \sigma_{yx} & \sigma_{\cos\theta x} & \sigma_{\sin\theta x} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{\cos\theta y} & \sigma_{\sin\theta y} \\ \sigma_{\cos\theta x} & \sigma_{\cos\theta y} & \sigma_{\cos\theta\cos\theta} & \sigma_{\cos\theta\sin\theta} \\ \sigma_{\sin\theta x} & \sigma_{\sin\theta y} & \sigma_{\sin\theta\cos\theta} & \sigma_{\sin\theta\sin\theta} \end{pmatrix}$$

$$= \begin{pmatrix} 0.5 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0.7 & 0 \\ 0 & 0 & 0 & 0.7 \end{pmatrix}$$

本稿では合計 324 個の場所領域となった．このうち 90 個の場所領域で画像が観測され，1 個の場所領域につき 3 から 32 枚の画像を観測した．本稿では合計 1905 枚の観測画像の物体認識結果から各領域の多項分布のパラメータ φ を学習した．

3.2.4 多項分布のパラメータの学習

推定した地図を元に自己位置推定を行い，観測された画像と自己位置を保存する．観測された画像は，推定した自己位置と各場所領域のガウス分布の中心を比較して最も近い場所領域に属するものとして以下のような式で近似した．

$$C_t = \arg\max_{C_t} p(x_t|\mu_{C_t}, \Sigma_{C_t}) \tag{14}$$

オフラインで，観測した画像を 1 枚ずつ Caffe で認識し，認識結果をカウントしたものを用いて場所領域ごとに多項分布のパラメータ φ を学習した．このときディリクレ分布のハイパーパラメータは $\alpha = 12$ とした．また本実験では物体認識結果として得られた確率を 10^2 倍した値を整数値に丸めたものをカウント数として用いた．

3.3 実験結果

3.3.1 提案手法によるパーティクルの状態と自己位置の誤差の検証

本研究の提案手法の有効性について検証した結果を以下に述べる．各試行に対して観測として用いる画像は一枚のみとした．その時のパーティクルの状態の変化の例を図 6 に示す．図の左側が制御情報と距離の観測情報のみを用いてパーティクルをある程度収束させた状態を示している．一方，図の右側がパーティクルをある程度収束させたあと提案手法により観測された画像の物体認識結果の情報を加えた時のパーティクルの変化を示している．各々の図に存在する赤い矢印はパーティクルの姿勢を示しており，矢印の方向はパーティクルが推定するロボットの方向である．実験結果の図より，制御情報と観測情報を用いて自己位置を推定したパーティクルの状態に，画像の物体認識結果の情報を加えた時，パーティクルが真の位置に収束していることがわかる．これより，提案手法により間違った位置を推定したパーティクルを減少させ，正しい位置に推定することができたと考えられる．

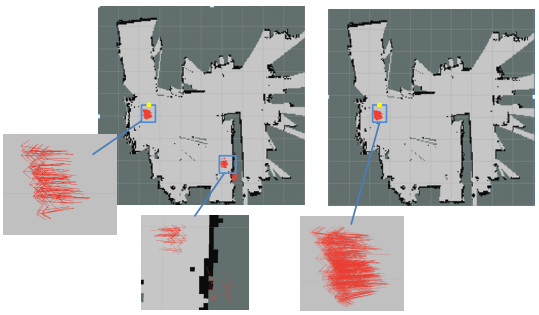

図 6: パーティクルの状態，左:物体認識結果を加える前，右:物体認識結果を加えた後．図中の黄色の点はロボットの真の位置を示す．

4. まとめと今後の課題

本稿では大域的な位置推定の誤りを減少させることを目的として MCL の制御情報，距離の計測情報に加えて，CNN による物体認識結果を用いることを提案した．

本稿では画像一枚を観測した場合のみで評価を行ったが，MCL は時間ごとに更新されるモデルであるため今後は時系列で観測された画像を用いて，位置推定に活用していく必要がある．また，多項分布の学習方法として地図の座標と方向により場所領域を決め，観測された画像を近くの場所領域の特徴として近似して学習を行ったが，観測された画像がどのクラスに属するかは距離だけで定められるものではなく，その場所で観測される画像と距離の近さによって分けられるものと考えられる．よって，得られた物体の認識結果とその時の自己位置を用いて領域を学習して実験を行うことが今後の課題である．

参考文献

[Krizhevsky 12] Krizhevsky, A., Sutskever, I., and Hinton, G. E.: ImageNet Classification with Deep Convolutional Neural Networks, in *Advances in neural information processing systems*, pp. 1097–1105 (2012)

[Thrun 05] Thrun, S., Burgard, W., and Fox, D.: *Probabilistic robotics*, MIT press (2005)

[岡谷 13] 岡谷貴之：Deep Learning (深層学習)(第 4 回) 画像認識のための深層学習, 人工知能学会誌, Vol. 28, No. 6, pp. 962–974 (2013)

[岡野 13] 岡野原大輔：Deep Learning (深層学習)(第 3 回) 大規模 Deep Learning (深層学習) の実現技術, 人工知能学会誌, Vol. 28, No. 5, pp. 785–792 (2013)

[谷口 14] 谷口彰, 吉崎陽紀, 稲邑哲也, 谷口忠大：自己位置と場所概念の同時推定に関する研究, システム制御情報学会論文誌, Vol. 27, No. 4, pp. 166–177 (2014)