

端点を持つ線から描くキャラクターの線画生成

学籍番号 20C1119 森田大雅

2023 年 10 月 21 日

概要

描画ロボットの研究において、輪郭 (エッジ) 抽出を行って鉛筆画やハッチング [1]、インクイラストなどの芸術表現に応用する研究 [2] や機械に手順を示し、そのとおりに描かせる研究 [3] が存在する。しかし、実際に「人が描くような描き方」を追求したものは少ないと感じた。そこで、今回はキャラクターの画像から線画を描く、そしてできるだけ人が描いたような描き方をする描画ロボットを作成する。

1 導入

描画ロボットにおいてプロッターやそれに似た機構で絵を描かせる研究や開発は多く見られるが、人の腕に似た機構であるロボットアームを用いて絵を描かせる研究はあまり存在しない。じつは現代の美術の人物などのデッサンにおいては輪郭を明確に描くやり方はあまり取らない。それは古典的なやり方で、昔の巨匠が実際そのように描くことは普通だった。今でも意図的な理由があって、輪郭を描くことである表現をしたい場合に使うことは確かにある。人に似たような描き方をさせたいという以上、輪郭やどこから描くなどの描き順に拘りたい所ではあるが、描き順のパターンを機械学習に任せて描かせることなども含めて行うのは難易度がかなり高い。そこで今回は 3 軸のロボットアームで、一枚の画像の輪郭を抽出し、人が描くような描き方を考えたい。線の経路を求める方法は現時点で 2 通り考えている。一方が左上から右下へ走査していき、線の画素を見つけたらたどるラスタスキャンの方法である。もう一方は端点を持つ線から描いていく方法である。ただしこちらはスタート地点をランダムにしており、場合によっては 1.1 の手順とは程遠くなってしまうことがある。しかし直感的にラスタスキャンより端点から描くほうが人が描いたように見えると考えたため、今回用いることにした。

1.1 人のような描き方の定義

本研究では [4] を参考に描き方の方針を進めている。顔のパーツ配置が定まりやすいという部分に焦点を当てた。描き順としては顎や髪、頭などの頭上部から描き、次に目や鼻、耳などの細部を描く。理由は全体を描

いてから、細部を描いた方が目や鼻の位置を定めやすいからである。

2 システム構成

このシステムは画像処理、経路探索、機械の動作の 3 つの構成である。

画像処理は線の経路を求めやすくするために行うもので、キャラクターの画像から平滑化、エッジ抽出、細線化などを行い線画を作成する。またその画像を元に経路を求め、そのときに取得した座標通りに機械に動かして描かせる。

2.1 画像処理の流れ

任意の一枚の画像を元に平滑化、エッジ抽出 (ラプラシアンフィルタ)、鮮鋭化 (4 近傍)、細線化、端点検出、経路を求め、そして取得した座標を元に機械に描かせる。

線の経路の求め方は、ある線の画素から隣に線の画素があるかを探して、移動してを繰り返すものとなっている。そのため、線を一本に単純化してある方が線をたどるのに容易であるため細線化を行う。

2.2 端点を持つ線から描く理由

端点を検出して描くするには 2 つの理由がある。線画を生成する過程で繋がっているはずの線が途切れてしまうというのが一つの理由である。それは平滑化、エッジ抽出、細線化処理を施すからである。

3 ロボットの機構

本研究で用いるロボットは3軸のマニピュレータロボットである。下の図1のようにアームの先端、姿勢の関節角を $\theta_1, \theta_2, \theta_3$ 、各リンクの長さ l_1, l_2, l_3, l_4 とし、逆運動学問題を解く。またリンク座標系を定める方法として、DH 記法 (Denavit-Hartenberg 記法) を用いる。

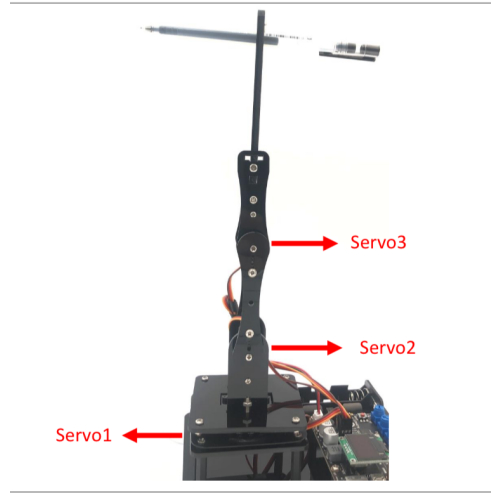


図 1: 実機

座標系 0 から 3 への座標変換行列

$${}^0T_3 = {}^0T_1 {}^1T_2 {}^2T_3$$

$$= \begin{pmatrix} C_1 C_{23} & -C_1 S_{23} & 0 & l_2 C_1 S_2 \\ S_1 C_{23} & -S_1 S_{23} & 0 & l_2 S_1 S_2 \\ -S_{23} & -C_{23} & -1 & l_2 C_2 + l_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

ただし、

$$\begin{cases} C_{23} = \cos\theta_2 \cos\theta_3 - \sin\theta_2 \sin\theta_3 \\ S_{23} = \sin\theta_2 \cos\theta_3 + \cos\theta_2 \sin\theta_3 \end{cases}$$

手先ベクトル

$$\begin{pmatrix} 1 & 0 & 0 & l_4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -l_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & l_4 \\ 0 & 1 & 0 & -l_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

より手先までの座標変換行列 0P_r が以下のように求まる。

$$\begin{pmatrix} C_1 C_{23} & -C_1 S_{23} & 0 & l_4 C_1 C_{23} + l_3 C_1 S_{23} + l_2 C_1 S_2 \\ S_1 C_{23} & -S_1 S_{23} & 0 & l_4 S_1 C_{23} + l_3 S_1 S_{23} + l_2 S_1 S_2 \\ -S_{23} & -C_{23} & -1 & -l_4 S_{23} + l_3 C_{23} + l_2 C_2 + l_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

そしてこの行列から手先の位置 x, y, z は以下のように求まる。

$$\begin{cases} x = C_1(l_4 C_{23} + l_3 S_{23} + l_2 S_2) & (3.1) \\ y = S_1(l_4 C_{23} + l_3 S_{23} + l_2 S_2) & (3.2) \\ z - l_1 = -l_4 S_{23} + l_3 C_{23} + l_2 C_2 & (3.3) \end{cases}$$

これらの逆運動学を解くと

$$\begin{cases} \theta_1 = \frac{1}{2} \cos^{-1} \left(\frac{x^2 - y^2}{x^2 + y^2} \right) \\ \theta_2 = \cos^{-1} \left(\frac{x^2 + y^2 + (z - l_1)^2 + l_2^2 - l_3^2 - l_4^2}{2l_2 \sqrt{x^2 + y^2 + (z - l_1)^2}} \right) + \tan^{-1} \left(\frac{\sqrt{x^2 + y^2}}{z - l_1} \right) \\ \theta_3 = \cos^{-1} \left(\frac{x^2 + y^2 + (z - l_1)^2 - l_4^2 - l_3^2 - l_2^2}{2l_2 \sqrt{l_3^2 + l_4^2}} \right) + \tan^{-1} \left(\frac{-l_4}{l_3} \right) \end{cases}$$

4 実験

今回は端点を持つ線から描く場合と、ラスタスキャンで線の経路を求めた場合を比較し、より人らしい描き方の方を用いる。

5 出力結果

以下に今回使用した実機で端点から描いたものと、ラスタスキャンで描いたものを載せる。

6 参考文献

- [1] Raspberry Pi を利用した肖像画描画ロボット : [Pankraz Piktograph]

2. [2] XDoG: An Xtended difference-of-Gaussians compendium
3. [3] ロボットによる描画行為の再現
4. [4] 出版: 主婦の友社「小河原智子の似顔絵入門」
5. 広瀬 茂男 著 機械工学選書 裳華房 ロボット工学-機械システムのベクトル解析-
6. 細田 耕著 実践ロボット制御-基礎から動力学まで-