

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 9 月 23 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

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。手順としては、最初に顎を描き、次に髪などの頭上部、そして首、

1. 顎
2. 頭上部 (髪、頭など)
3. 首
4. 目、鼻、耳
5. その他 (胴体や服など)

これは顎や髪、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という順序である。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

このシステムは主に 3 つから構成されている。1 つが画像から輪郭を抽出する。2 つが抽出した画像から線の経路を求める。3 つが線の経路をもとに実機を動かして描かせる。

2.1 処理の流れ

細かな処理の流れは以下の通りである。

1. 平滑化
2. エッジ抽出
3. 細線化
4. 端点の検出
5. 経路を求め、移動すべき座標を取得
6. 取得した座標をもとに機械に描かせる

2.2 エッジ抽出

輪郭抽出にはエッジ抽出が一般的に行われる。ソーベルフィルタやケニーのエッジ検出なども検討したが、実際に試したところラプラシアンフィルタをエッジ抽出に用いることにした。

2.3 細線化

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。

2.4 端点を持つ線から描く理由

端点を検出して描くのに 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。

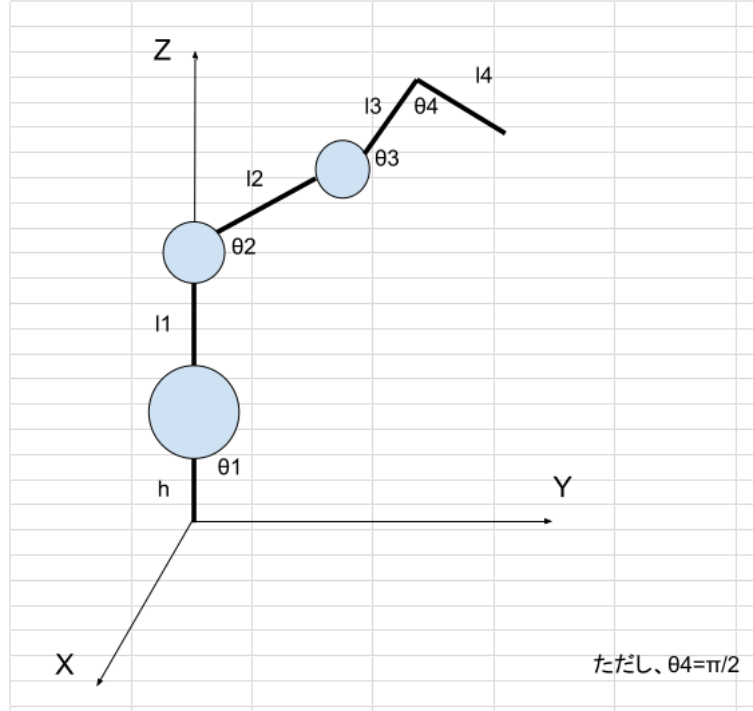


図 1: リンク座標系

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

$$\begin{cases} \theta_1 = \frac{1}{2} \cos^{-1} \left(\frac{x^2 - y^2}{x^2 + y^2} \right) \\ \theta_3 = \cos^{-1} \left(\frac{1}{\sqrt{a^2 + b^2}} \right) + \cos^{-1} \left(\frac{b}{\sqrt{a^2 + b^2}} \right) \\ \theta_2 = \cos^{-1} \left(\frac{1}{\sqrt{c^2 + d^2}} \right) + \cos^{-1} \left(\frac{d}{\sqrt{c^2 + d^2}} \right) \end{cases}$$

ただし、

$$\left(\begin{array}{l} a = \frac{2l_2}{l_4^2 + l_3^2 + l_2^2 - \{x^2 + y^2 + (z - l_1)^2\}} l_4 \\ b = \frac{2l_2}{l_4^2 + l_3^2 + l_2^2 - \{x^2 + y^2 + (z - l_1)^2\}} l_3 \\ c = \frac{l_2 S_3 - l_4}{C_3(z - l_1) + S_3\left(\frac{x}{C_1}\right)} \\ d = \frac{l_2 S_3 + l_3}{C_3(z - l_1) + S_3\left(\frac{x}{C_1}\right)} \end{array} \right)$$

4 参考文献

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