

目次

はじめに	3
第1章 研究背景	3
1.1 身体知ということば	3
1.2 用語の定義づけ	3
1.3 本論文における言葉遣い	4
1.4 本論文の構成	5
第Ⅰ部 身体運動の創造的学びの物語	2
第1章 研究背景	3
1.1 背景	3
1.2 マブラヴオルタネイティブ	3
1.3 色の知覚	4
第Ⅱ部 身体の「表情」感得を促すアプリの制作と、その実践	5
第2章 アプリのしくみ	6
2.1 どういうアプリか？	6
2.2 システム構成	7
2.3 アプリの開発環境	9
2.4 データベース概要	10
2.5 アプリの機能とユーザインタフェース	10
参考文献	15

付録	17
第 3 章 アプリについての付録	17
3.1 最近傍マーカの計算法	17
3.2 開発にもちいた JavaScript パッケージ	19

身体運動の創造的学びの一人称研究

堀内隆仁

2023 年 12 月 9 日

はじめに

第 1 章

研究背景

本研究は第一部第二部からなる。両方の部に通底する風呂敷を広げておこう。なんやかんあ y。

1.1 身体知ということば

いっぱい使われ方がある。古くはギリシャ哲学や東洋哲学の流れにも、私たちの知性の本質を身体に求めようとする流れはあった。ひとつの大きな転換点は、1980 年代ころだろう。当時エキスパートシステムによって栄えていた人工知能研究が、おおきな壁にぶち当たった。フレーム問題である。そのころ、エキスパートシステムにはない「身体」という存在こそが、人間知性の本質ではないかということで、ひとの知能・知性を探究するさまざまな学問領域で、身体の知の探究が活発になった。

1.2 用語の定義づけ

そういうわけから、本研究では、さまざまな用語をもちいることにする。本性では、学際研究としてののにあるということを強くして本章では、それらを道具立てとして、定義しておく。

身体立てる

身体知のもっとも広い意味での言い換えである。本研究での身体知とは、私たちがものごとを認知するとは、いまだ不定の X を、「なにか」として身体立てることであると定義するのだ。

流転する

あるもの X「変化のなかにある」ことを、「X が流転する」と表現する。あるもの A が A として在ることができるのは、たえまなく「変化」のなかに晒されているのだ。。よおお。「変化」と表現すると、ものが A として盤石にあって、

1.3 本論文における言葉遣い

本論文では、必要があれば、積極的に「メタファ」や「」をももちいる。なぜか。知を身体知とみる本研究では、論述する（議論する）という人間の知も、身体知とみなすことになる。身体知としての論述は、「（読者や著者の）身体から地続き」になるように展開すべきものであると考える。メタファは、身体とつながった言葉の表現方法である（尼崎）。科学論文では、メタファの使用は御法度とされている。メタファは「詩的」であり、それは「論理的」であることと相反するという考え方にもとづくのだろう。だが、それは頭知パースペクティブからみた結果にすぎないと著者は考える。

読者に「身体でわかる」ということを促すように説明するには、論理的説明がいつでも最適とは限らない。むしろ、すべてを論理的に説明しようとしすぎると、身体性が欠落した説明にもなりかねない。本論文がそうになってしまえば、「主張する形式（文章の書き方）」が「主張している内容（知は身体知である）」と自己矛盾していることになる。

このうえであらためて「論理的」という言葉を見つめ直してみると、「ひとつひとつのことがらを地続きにつなげる」という意味があることに気づく。メタファの活用は、「対象のことがらと身体とを地続きにする」のならば、それは「論理的」であることとも密接に関係している（詩的であることと論理的であることは相反というわけでもない、ということになる）。くわえるに、「が高い」とか「ー」とか、私たちがあまり疑いなく論理的表現として使っている言葉も、もともとはメタファだったという事実もある（リクール）。*1。

以上の理由から、本研究では、論理のなかに適切箇所ではメタファを組み入れることで*2、読者に「身体でわかる」ことを促したい。

知の原理がちがうのだから。

また同様の考え方にもとづき、必要があれば「冗長な説明」もいとわない。現象学的なことこれは、システムのしくみを説明する章でも、（丁寧すぎる説明は、なるべく脚注にまわすが）

*1 こういう表現を、リクールは「死んだ隠喩」と呼んでいる

*2 メタファをもちいているときは、そうであると明示する

1.4 本論文の構成

参考文献

付録

第 3 章

アプリについての付録

3.1 最近傍マーカの計算法

コンピュータグラフィックスの処理課程（グラフィックスパイプラインもしくはレンダリングパイプライン）は、以下の流れである。たとえば、ある 3 点があるとして説明する。

1. モデリング：世界座標系において、対象の 3 次元位置座標を決める
2. 視点変換：対象をカメラからの眺めに変換（世界座標系からカメラ座標系への並行移動）
3. （カリング）
4. クリッピング
5. （ラスタライズ）
6. 画面に出力

なお、ベクトルはボールド体で表記する。空間にある 2 つのマーカの 3 次元位置座標 \mathbf{p}, \mathbf{q} と、画面上のマウスクーソルの 2 次元座標 \mathbf{m} の「距離」 d を算出する。マーカの位置をそれぞれ、

$$\mathbf{p} = \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix}, \mathbf{q} = \begin{pmatrix} q_x \\ q_y \\ q_z \end{pmatrix}, \mathbf{m} = \begin{pmatrix} m_x \\ m_y \\ m_z \end{pmatrix}$$

モデル行列は今回は関係ない。カメラの外部パラメータと内部パラメータによって、変換ができる。

カメラの外部パラメータは、以下 3 つからなる。

- 世界座標系におけるカメラの位置（3 次元ベクトル）

- 世界座標系におけるカメラの注視点 (3次元ベクトル)
- 世界座標系におけるカメラの上方向 (3次元ベクトル)

これら3つの情報は、1つの「ビュー行列」にまとめて表現できる。

$$\mathbf{V} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

カメラの内部パラメータは、以下4つ^{*1}からなる。

- 焦点距離
- アスペクト比
- ニアクリップとファークリップ平面の距離
- 視野 (Field of View, FoV)

これら4つの情報は、1つの「プロジェクション行列」にまとめて表現できる。

$$\mathbf{P} = \begin{pmatrix} \frac{1}{\tan(\text{FOV}/2) \times \text{aspect}} & 0 & 0 & 0 \\ 0 & \frac{1}{\tan(\text{FOV}/2)} & 0 & 0 \\ 0 & 0 & \frac{f+n}{f-n} & \frac{2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

したがって、対象点の3次元ベクトルに、ビュー行列をかけてからプロジェクション行列をかけることで、変換後の画面座標系で表された2次元ベクトル $\mathbf{p}' = \begin{pmatrix} p'_x \\ p'_y \end{pmatrix}$ が得られる。

$$\mathbf{p}' = \mathbf{P}\mathbf{V}\mathbf{p} = \begin{pmatrix} \frac{1}{\tan(\text{FoV}/2) \times \text{aspect}} & 0 & 0 & 0 \\ 0 & \frac{1}{\tan(\text{FoV}/2)} & 0 & 0 \\ 0 & 0 & \frac{f+n}{f-n} & \frac{2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} p_x \\ p_y \\ p_z \\ 0 \end{pmatrix}$$

すべてのマーカー (n 個) に対して、これらの画面上2次元座標を求める。すべてのマーカーの画面上2次元座標とマウスカーソルとの距離をそれぞれ求める (d_1, d_2, \dots, d_n)。あ

^{*1} これら4つの情報もちいて、図のように、カメラからみた世界全体から、ある「四角錐台」の領域を一意に定めている。この四角錐台の領域のみ、「カメラに映り込む世界」として切り取るわけだ。なぜ直方体ではなく四角錐台なのか？それは「カメラに近ければ近いほど大きく写り、遠ければ遠いほど小さく映る」ということを反映するためである。このあと四角錐台を、2つの底面を押しつぶすように、二次元平面へと圧縮するが、それで反映される。

らはじめ画面 2 次元平面上でマウスカーソルを中心とした一定の半径 r 内にある点群を、「カーソル近傍マーカ」とするようしておき、 $r > d$ なる点群を求める。

「カーソル近傍マーカ」群のうちもっとも「カメラからみて手前」にあるものを、「最近傍マーカ」とする。

3.2 開発にもちいた JavaScript パッケージ

あかーいテキスト feiji 以下が npm パッケージ一覧である。

```
function hello(){  
  console.log('hello')  
}
```