進捗報告

B4 田川幸汰

1 概要

課題 2 と課題 3 についての進捗報告を行う。
課題 2 では、全方位画像を撮影して, その画像から指定した方向の透視投影画像を生成する。この課題を通して正距円筒画像から透視絵投影画像の生成方法について学習する。
課題 3 では bash のスクリプトと課題 2 で作成したプログラムを用いて, 水平方向の視線を $-\pi$ から π まで 10 度刻みで変化させた透視投影画像を生成する。この課題を通してシェルスクリプトの使い方について学習する。

2 課題 2. 全方位画像から透視投影画像生成

実装方法と理論について、以下の節でまとめる。

2.1 課題 2 の実装方法

以下に全方位画像から透視投影画像を生成する方法をまとめる [1]。詳しい理論については以降の節で説明する。

- 入力... 正距円筒画像 I_e、水平方向及び垂直方向の画角 Θ, Φ、視線角度 $\theta_{eye}, \phi_{eye}, \psi_{eye}$
- 出力... 透視投影画像 I_p
- アルゴリズム

1. 出力画像サイズ W_p , H_p を計算する。
2. 水平方向、垂直方向の画素間の長さ $\Delta\ x, \Delta\ y$ を計算する。
3. 透視投影画像 I_p の画素 (u_p, v_p) から、3 次元の視線ベクトル $\boldsymbol{x}$ を計算する。
4. 視線ベクトル $\boldsymbol{x}$ を回転行列 $R(\theta_{eye}, \phi_{eye})$ 、$R(\psi_{eye})$ により回転する。
5. 4. で求めた視線ベクトル $\boldsymbol{x}$ から、(θ, ϕ) を計算する。
6. 正距円筒画像 I_e 上の対応する画素 (u_e, v_e) を計算して、その画素値を透視投影画像 I_p の画素 (u_p, v_p) に割り当てる。

2.2 画素間の長さ

水平、垂直方向の画角 Θ, Φ と画像サイズ W_p, H_p が与えられているとき、画像面までの距離を $1(z = 1)$ とすると、水平方向、垂直方向の画素間の長さ $\Delta\ x, \Delta\ y$ は次のように与えられる。

$$\Delta\ x = 2\tan(\Theta/2)/W_p \tag{1}$$

$$\Delta\ y = 2\tan(\Phi/2)/H_p \tag{2}$$

2.3 視線ベクトルの計算

透視投影画像の画素 (u_p, v_p) とすると、式 (1)、式 (2) で求めた $\Delta\ x, \Delta\ y$ を用いて 3 次元の視線ベクトル $\boldsymbol{x}$ は次のように与えられる。

$$\boldsymbol{x} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} (u_p - W_p/2)\ \Delta\ x \\ (v_p - H_p/2)\ \Delta\ y \\ 1 \end{pmatrix} \tag{3}$$

2.4 透視投影画像への投影

正距円筒画像の一部を透視画像の画像面に投影することを考える。正距円筒画像は球面上に投影した画像とも考えられるので、透視投影の画像面の画素に対応する正距円筒画像の角度情報が得られれば、正距円筒画像の画素を透視投影画像に投影できる。

透視投影面上の座標と角度 (θ, ϕ, ψ) の関係を下の図 1 に示す。

上記の定義より、式 (3) で求めた透視投影面上の座標 (x, y, z) から、角度 (θ, ϕ) は次のように与えられる。

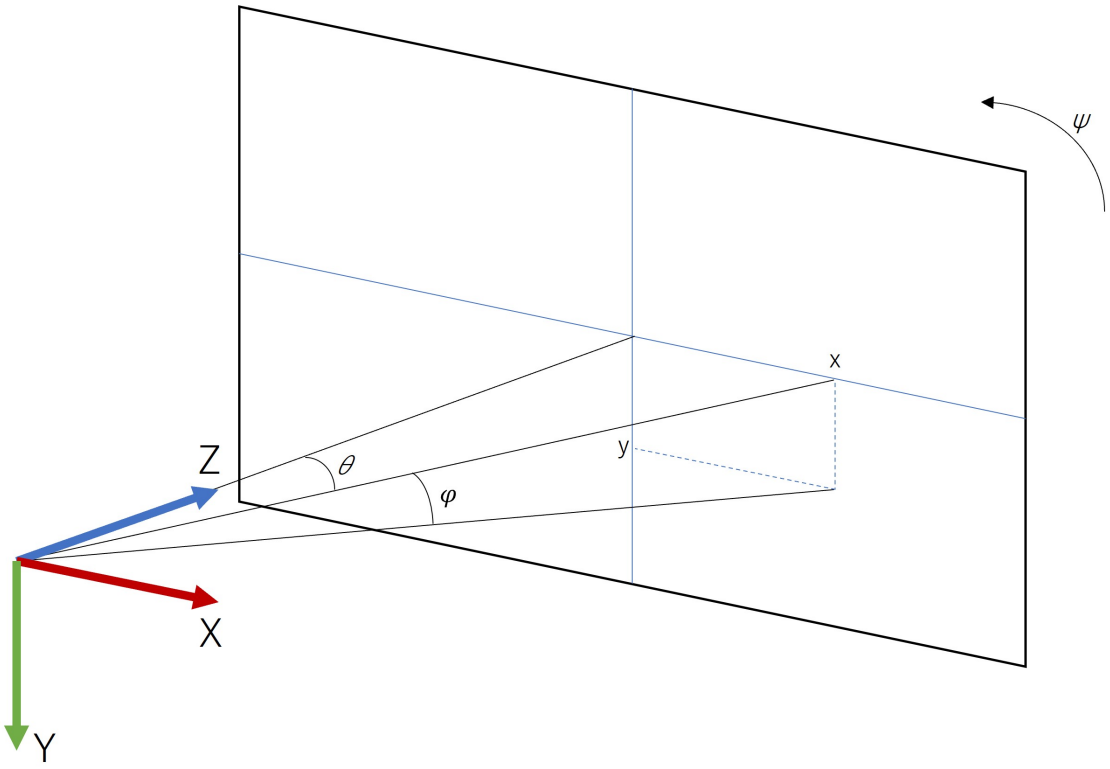

図 1: 角度の定義

$$\theta = \tan^{-1}\frac{x}{z} \tag{4}$$

$$\phi = -\tan^{-1}\frac{y}{\sqrt{x^2 + z^2}} \tag{5}$$

ただし、X 軸周りの回転は Y 軸から Z 軸方向への回転が正の回転であることを考えて、ϕ の符号は反転している。

2.5 360 度画像の座標系変換

360 度カメラで撮影した画像は、正距円筒画像として保存される。これはシーンを球面に投影したものを緯度、軽度を画像の縦方向と横方向に対応付けて矩形の画像として再現したものである。

カメラ画像の Z 軸 (光軸) が正距円筒画像の中心を通るとき、画像中心の θ-ϕ 座標系での座標が $(\theta, \phi) = (0, 0)$ となると考えると、画像の左上を原点として、右方向に u 軸、下方向に v 軸をとる座標系との関係は次のように表せる。

$$(\theta, \phi) = ((u - \frac{W}{2})\frac{2\pi}{W}, (\frac{H}{2} - v)\frac{\pi}{H}) \tag{6}$$

$$(u, v) = ((\theta + \pi)\frac{W}{2\pi}, (\frac{\pi}{2} - \phi)\frac{H}{\pi}) \tag{7}$$

ここで、W 及び H は入力画像の幅及び高さである。

2.6 画像サイズの決定

入力画像の解像度に合わせて出力画像を生成する場合、画角に応じた適切な画像サイズを計算する必要がある。入力画像の横サイズ W_e 及び縦サイズ H_e 、出力画像の水平、垂直方向の画角 (Θ, Φ) が与えられているとき、画像の横サイズ W_p 及び縦サイズ H_p は次のように与えられる。

$$W_p = 2\tan(\frac{\Theta}{2})\frac{W_e}{2\pi} \tag{8}$$

$$H_p = 2\tan(\frac{\Phi}{2})\frac{H_e}{\pi} \tag{9}$$

2.7 視点移動

画像面の回転がない場合、視点方向 $\boldsymbol{v} = (0, 0, 1)^\top$ である。視点の回転を行う場合、まず X 軸周りに ϕ_{eye} だけ回転したあと、Y 軸周りに θ_{eye} だけ回転すれば良い。したがって、そのための回転行列は次のように与えられる。

$$R(\theta_{eye}, \phi_{eye}) = \begin{pmatrix} \cos\theta_{eye} & 0 & \sin\theta_{eye} \\ 0 & 1 & 0 \\ -\sin\theta_{eye} & 0 & \cos\theta_{eye} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi_{eye} & -\sin\phi_{eye} \\ 0 & \sin\phi_{eye} & \cos\phi_{eye} \end{pmatrix} \tag{10}$$

また、光軸周りに ψ_{eye} だけ画像面を回転する場合、視点方向 $\boldsymbol{v}$ を $R(\theta_{eye}, \phi_{eye})$ で回転したあとのベクトルを新たに回転軸として回転を行えばよい。新たな回転軸 $\boldsymbol{l}$ 周りの回転行列は、ロドリゲスの定理より次のように与えられる。

$$R(\psi_{eye}) = \begin{pmatrix} l_x^2(1-\cos\psi_{eye}) + \cos\psi_{eye} & l_x l_y(1-\cos\psi_{eye}) - l_z\sin\psi_{eye} & l_z l_x(1-\cos\psi_{eye}) + l_y\sin\psi_{eye} \\ l_x l_y(1-\cos\psi_{eye}) + l_z\sin\psi_{eye} & l_y^2(1-\cos\psi_{eye}) + \cos\psi_{eye} & l_y l_z(1-\cos\psi_{eye}) - l_x\sin\psi_{eye} \\ l_z l_x(1-\cos\psi_{eye}) - l_y\sin\psi_{eye} & l_y l_z(1-\cos\psi_{eye}) + l_x\sin\psi_{eye} & l_z^2(1-\cos\psi_{eye}) + \cos\psi_{eye} \end{pmatrix} \tag{11}$$

なお、$\boldsymbol{l} = (l_x, l_y, l_z)^\top = R(\theta_{eye}, \phi_{eye})\boldsymbol{v}$ である。
これらの回転行列と、式 (3) で求めた透視投影面上の視線ベクトル $\boldsymbol{x}$ を用いて、視点移動後の視線ベクトル $\boldsymbol{x}'$ は次のように与えられる。

$$\boldsymbol{x}' = R(\psi_{eye})R(\theta_{eye}, \phi_{eye})\boldsymbol{x} \tag{12}$$

2.8 課題 2 の実装

2.8.1 入力

- 全方位画像
  図 1 は課題 2 の実行に用いた全方位画像である。

- 出力画像の画角
  Θ(横方向) = 45 ° 、Φ(縦方向) = 45 °

- 視線ベクトル
  θ(横方向) = -45 ° 、ϕ(縦方向) = -20 ° 、ψ(光軸方向) = 45 °

図 2: 全方位画像

2.8.2 出力

図 2 の全方位画像から、図 3 の透視投影画像が出力された。
画素が荒くなってしまったが、視線ベクトルを指定することで狙った方角の透視投影画像が出力できていることがわかる。

図 3: 透視投影画像

3 課題 3.bash を用いた画像生成

実装方法について、以下の節でまとめる。

3.1 プログラムの引数の変更

bash で扱いやすいようにプログラムの入力を変更する。ここでは、プログラムの引数として、水平方向の視線ベクトルを与える。

3.2 シェルスクリプトの作成

ソースコード 1 は視線角度を $-\pi$ から π まで 10 度ずつずらし python プログラムを実行するシェルスクリプトである。

Listing 1: 連続な透視投影画像を表示

```
#!/bin/bash

for i in $(seq 0 36)
do
  python3 moveviewpoint.py kadai2.jpg input.csv $i
done
```

また、シェルスクリプトの作成に合わせて、透視投影画像を表示する python プログラムも一部変更した。

3.3 課題 3 の実行

3.3.1 入力

入力画像、画角は課題 2 と同じ。高軸方向の視線ベクトルは $\psi = 0$ とした。

3.3.2 出力

全 37 枚の画像が表示された。その中で一部の画像を下に示す。図 4 と図 5 を見ると、右方向に平行移動していることがわかる。また、図 4 と図 7 は同じになった。これにより、透視投影画像の視点が水平方向に一周していることがわかる。

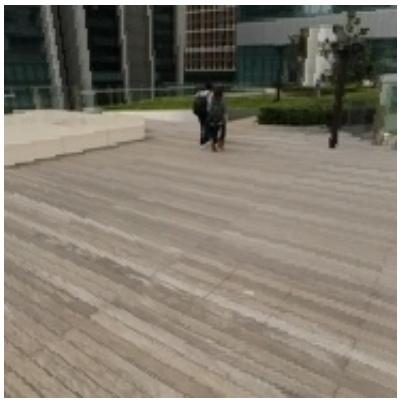

図 4: θ(横方向) = -180 °

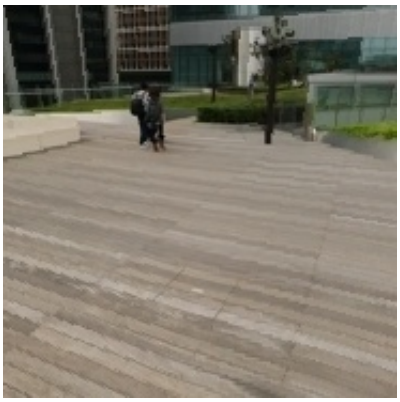

図 5: θ(横方向) = -170 °

図 6: θ(横方向) = 0 °

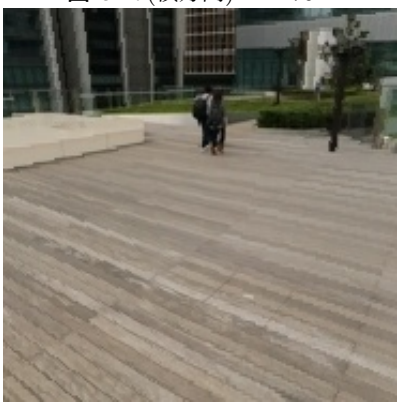

図 7: θ(横方向) = 180 °

参考文献

[1] 菅谷保之, 360 度画像からの透視投影画像の生成 (3), 4 5p, 最終更新日 2023 年 4 月 5 日.