



テクスチャの修正と自己位置推定

M2 田川幸汰

1 概要

3次元モデルのテクスチャに変更を施し、再現精度を高めた。主な変更点は以下の2点である。

1. 側面テクスチャと床面テクスチャを四角形形状のテクスチャに変更
2. テクスチャを裏側から取得しないように修正

2 テクスチャ形状の変更

三角形のテクスチャを使用する場合、テクスチャ割り当て時にずれが生じやすく、見た目がちぐはぐになって再現精度が低下する¹。

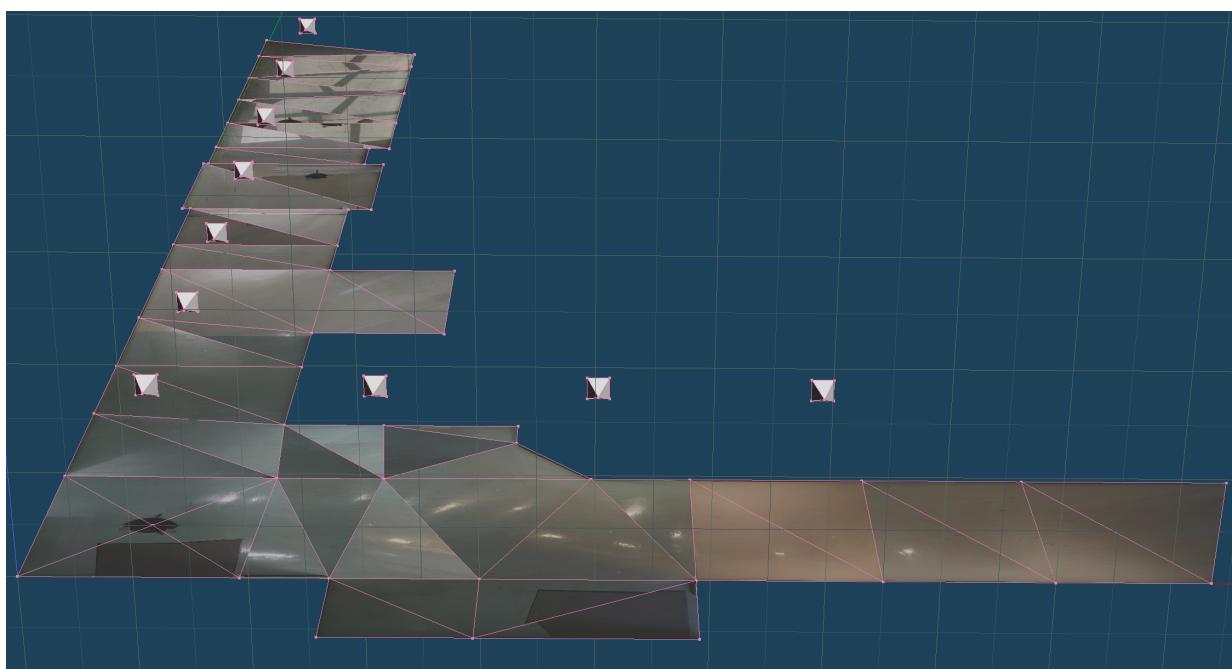


図 1: 3次元モデル

また、三角形で構成することで面の数が多くなり、それに伴い必要なテクスチャの枚数も増加してしまう。さらに、側面と床面でテクスチャの形状が異なると、管理が煩雑になる可能性が高くなる。

これらの問題を解決するために、床面についてもテクスチャ形状を四角形に統一する。これにより、再現精度の向上やテクスチャ枚数の削減、管理の簡素化が期待できる。

ただし、床面のテクスチャは側面と異なり、形状が一定ではない。そのため、テクスチャの四隅の座標をもとにUV座標を算出し、射影変換に用いる必要がある。



2.1 UV 座標の導出

UV 座標とはテクスチャを3次元オブジェクトの表面に貼り付けるための座標系である。同一平面上に存在する3頂点 $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2$ に対して、UV 座標を構成する U 軸は以下のように定義する。

$$\mathbf{u} = \frac{\mathbf{p}_1 - \mathbf{p}_0}{|\mathbf{p}_1 - \mathbf{p}_0|} \quad (1)$$

また、V 軸は U 軸と 3 頂点で構成される平面の法線ベクトル \mathbf{n} の外積で定義する。

$$\mathbf{n} = \frac{(\mathbf{p}_1 - \mathbf{p}_0) \times (\mathbf{p}_2 - \mathbf{p}_0)}{|(\mathbf{p}_1 - \mathbf{p}_0) \times (\mathbf{p}_2 - \mathbf{p}_0)|} \quad (2)$$

$$\mathbf{v} = \mathbf{u} \times \mathbf{n} \quad (3)$$

\mathbf{p}_0 を UV 座標系の原点として、頂点 \mathbf{p}_i を UV 座標に投影する。

$$\mathbf{d}_i = \mathbf{p}_i - \mathbf{p}_0 \quad (4)$$

$$u_i = \mathbf{d}_i \cdot \mathbf{u} \quad (5)$$

$$v_i = \mathbf{d}_i \cdot \mathbf{v} \quad (6)$$

$$(7)$$

また、UV 座標は $[0, 1]$ の範囲のため、以下のようにシフトとスケーリングの処理を行った。

$$\mathbf{uv}'_i = [u_i, v_i] - \min_j([u_j, v_j]) \quad (8)$$

$$\mathbf{uv}''_i = \mathbf{uv}'_i \cdot \frac{1}{\max_j([u_j, v_j])} \quad (9)$$

$$(10)$$

計算された UV 座標は、生成されたテクスチャのひずみをなくすために用いる。ひずみをなくすための射影変換行列を求めるために、opencv の getPerspectiveTransform メソッドを用いる。また、射影変換行列を用いた画像変換にはopencv の warpPerspective メソッドを用いる。

2.2 生成されたテクスチャ

生成された側面テクスチャ及び床面テクスチャを図 2 に示す。

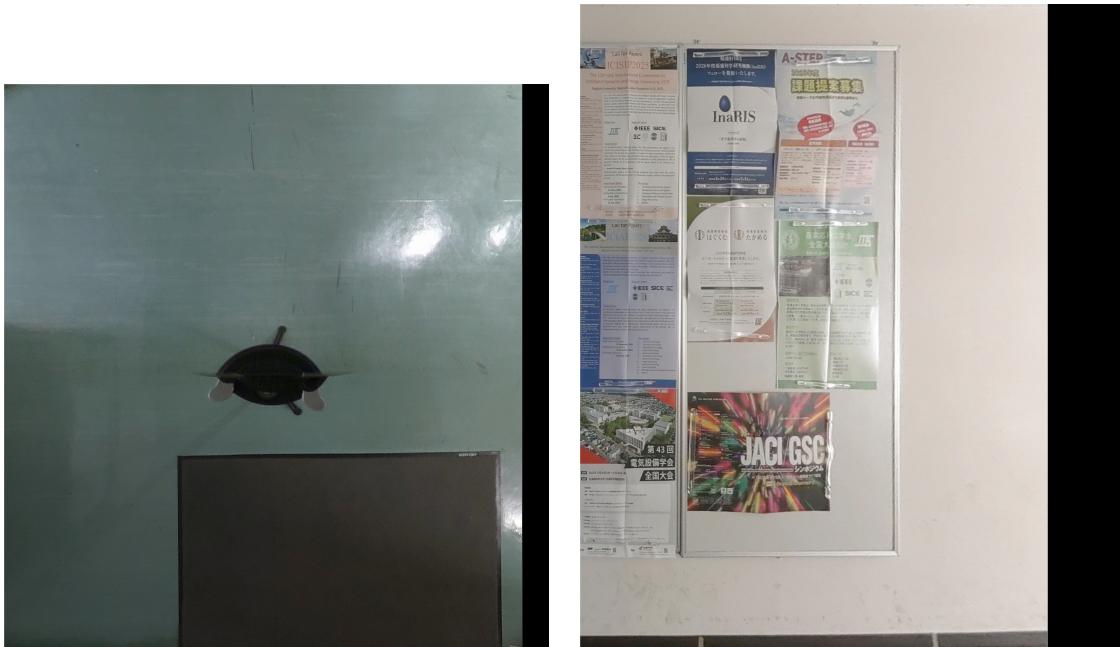


図 2: テクスチャ画像



床面、側面ともに正しく生成することができている。

3 今後の計画

今後の研究計画を以下に示す。

1. 拡張した3次元モデルの側面部のテクスチャ割り当て
2. 自己位置推定機能を応用した実用的なアプリケーションの開発
 - 生成された3次元モデルと自己位置推定を組み合わせ、目的地までのルートを提示するナビゲーションシステムを構築