

認識で欠かせない! 実用画像補正テクニック

第3回
最終回

処理が軽くてプログラムがシンプル!
決定版!? 変換アルゴリズム

外村 元伸

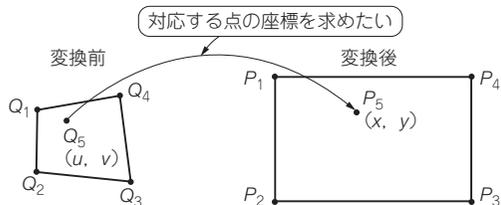


図1 連載の目的…画像認識の基本的な前処理! 斜め画像を正面画像にしたい

被写体を斜めから撮影すると、画像は斜めにひずんでいます。本連載は、この斜め画像を正面から見た画像に変換する方法を解説することが目的です(図1)。第1回、第2回では図2(a)(b)に示す方法を紹介してきました。今回は第2回の方法をさらに進化させて、軽くてシンプルに変換を行ってみます[図2(c)、写真1]。

● これまでの流れ

連載第1回は、変換前の画像、変換後の画像、両方の位置対応がとれる4点(例:ナンバープレートの隅)を選び、両方の各位置を結びつける射影変換行列を、方程式によって解く方法を解説しました[図2(a)]。

ところが第1回の方程式は計算的に複雑かつ、数値

変換式が複雑になってしまう

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \frac{\begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}}{h_{31}x + h_{32}y + h_{33}}$$

(a) 第1回…射影変換法を用いたが計算量が多い

$$\frac{Q'_1 P'_0}{Q'_1 P'_0} = \frac{t \cdot P_1 P_2 \cdot P'_1 P'_0 \cdot P'_0 P'_2}{Q_1 P_2 \cdot P'_1 P'_2 - t \cdot P_1 P_2 \cdot P'_1 P'_0}$$

(b) 第2回…変換式がシンプルになったため計算量が減った

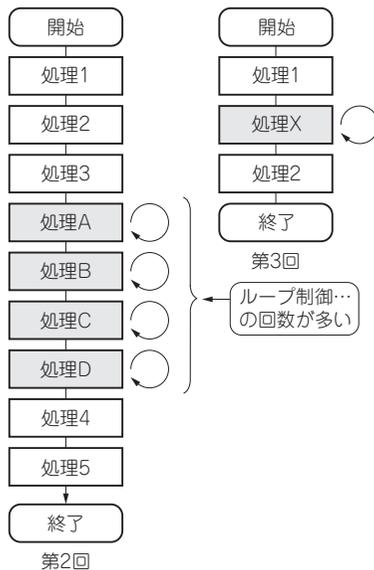
図2 第1回で生じた問題を第2回、第3回と改良してきた



写真1 面積複比アルゴリズムによる斜め撮影ナンバープレートの正面画への復元

解法的に不安定な場合があります。そこで第2回では「複比」と呼ばれる方程式を用いて、変換後の値を求めました。式がシンプルになったぶん、CPUへの負担を減らすことができます[図2(b)]。

しかし第2回の方法は、距離複比の関係を描画点ご



(c) 第3回…数式やプログラムが第2回と比べてとてもシンプルで済む[計算量は(b)と同様]