

ご購入はこちら

ゆがみや欠陥の補正

土井 伸洋

3-1, パース変換

● できること

▶ 画像のひずみを補正する

撮影対象物を真正面からとらえられない場合、例えば大きな建築物を見上げて撮影したり、机の上に置かれた書類を斜めから撮影するような場合、対象物がひずんだ画像となります。これを補正する処理がパース変換です。

▶ 画像をひずませる

プロジェクタの台形補正やプロジェクション・マッピングにおいては、あらかじめひずませた画像を投影することで、投影そのものによるひずみを打ち消し、希望する画像を対象物上に展開できます(図1)。

● プログラムと実行結果

ここでは机の上に置いた冊子をパース変換により補正する例を示します。プログラムをリスト1に示します。この処理には点の対応関係から変換行列を求める `cv2.getPerspectiveTransform()` 関数と、得られた変換行列を画像に適用する `cv2.warpPerspective()` 関数の2つの関数を使います。パース変換を行う際には、補正前および補正後の対応する4頂点の座標が必要になります。今回はプログラム上で直接数値を指定しましたが、多くの画像ソフトウェアでは、GUI画面上で頂点を選びリアルタイムで変形できる機能が提供されています。また、あらかじめ寸法分かっている書類の四隅にマーカを付けておけば、マーカを検出しパースを自動で補正することもできます。

実行結果を図2に示します。パース変換後の画像では、多少のひずみはあるものの、冊子を正面から撮影したものとほぼ変わらない画像が得られています。

● 深層学習での使いどころ

パース変換による補正は、写真に写った文字や数値を認識する場合の前処理としてよく使われます。撮影



図1 プロジェクションマッピングでは画像をひずませてさいころに投影した

リスト1 画像のパース変換

```
# 画像の読み込み
src_image = cv2.imread("pers_sample.png")
src_image = cv2.cvtColor(src_image,
                          cv2.COLOR_BGR2RGB)

# パース変換
points_before = np.array([[225, 60], [985, 60],
                           [1170, 860], [50, 865]], np.float32)
points_after = np.array([[20, 20], [580, 20],
                          [580, 820], [20, 820]], np.float32)
# before-after の点の対応関係から、変換行列を計算する
matrix = cv2.getPerspectiveTransform(points_before,
                                     points_after)

# 画像に変換行列を適用する
# M: 変換行列
# dsize: 出力画像サイズ (width,height)
image = cv2.warpPerspective(src_image, M=matrix,
                             dsize=(600, 840))
```



(a) 入力画像

(b) 出力画像

図2 リスト1 (画像のパース変換) の実行結果