

付録: 本実験における OpenCV の使用例

詳細は [OpenCV のリファレンスマニュアル](#)などを参考にする事。

- ① **`z = Depth.at<unsigned short>(i, j);`**
Depth は unsigned short 型の要素を持つ OpenCV の行列型である。この例では, Depth の第 i, j 要素を z に代入している ($0 \leq i < 480, 0 \leq j < 640$)。
- ② **`bgrImage.at<Vec3b>(i, j) = Vec3b(127, 0, 0);`**
行列型のカラー画像 bgrImage の第 i, j 要素に暗い青 ((青, 緑, 赤)=(127, 0, 0)) を代入する ($0 \leq i < 480, 0 \leq j < 640$)。
- ③ **`Mat color(Size(640, 480), CV_8UC3, Scalar(0,0,255));`**
縦 480×横 640 のカラー画像 color を行列型で確保する。この例では, 全画素が赤 ((青, 緑, 赤)=(0, 0, 255)) に初期化される。
- ④ **`imwrite("filename.png", A);`**
行列 A が PNG 形式の画像として保存される。
- ⑤ **`circle(A, Point(j, i), 8, Scalar(255,0,0), -1);`**
// circle(画像, 円の中心座標, 半径, 色, 線太さ);
行列型のカラー画像 A に円を描き込む。この例では, 円の中心は画素 (i, j) ($0 \leq i < 480, 0 \leq j < 640$), 半径は 8, 色は青, 線の太さは -1 (塗りつぶし) である。
- ⑥ **`Point3f pa, pb, pc, pd;`**
`pa.x = 0.1; pa.y = 0.2; pa.z = 0.3;`
`pb.x = 0.8; pb.y = 0.5; pb.z = 1.2;`
`pc = pb - pa; pd = pb + ba;`
3 次元ベクトルの成分に代入し, 和と差を計算できる。Point3f 型は, float 型のメンバ x, y, z を持つ。その他, Vec3f 型など, ベクトルの演算が定義されている型もある。 [リファレンスマニュアルの基本構造体](#)を参照のこと。
- ⑦ **`putText(A, "abc", Point(300,400), FONT_HERSHEY_SIMPLEX,`**
`4.0, Scalar(0,255,0), 12, CV_AA);`
//putText(画像, テキスト, 位置 (左下), 字種, スケール, 色, 線太さ, 種類); 行列型のカラー画像 A の位置 (300, 400) に緑色の文字列 "abc" を描き込む。