『OpenCV による画像処理入門』第 1 刷正誤表

この度は，標記書籍をお買い求めいただき誠にありがとうございました。
標記書籍に誤りがありました。訂正し，深くお詫び申し上げます。

ページ数	行数	位置	誤	正
35	5行目		image.channels	image.channels()
		図3.16		
38	下から4行目		img_dst = cv2.flip(img_src, flipCode = 0); # 垂直反転	img_dst = cv2.flip(img_src, flipCode = 0) # 垂直反転
39	17行目			
39	1行目追加		19 cv2.waitKey(0) # キー入力待ち 20 cv2.destroyAllWindows()	19 cv2.imwrite(img_dst) # 処理結果の保存 20 cv2.waitKey(0) # キー入力待ち 21 cv2.destroyAllWindows()
46	17行目以降		1 # 複数色チャンネルの分割 2 Vector<Mat> img_bgr(3); 3 split(img_src,img_bgr); 4 # 青→赤，緑→青，赤→緑に変更 5 merge((img_bgr[2],img_bgr[0],img_bgr[1]),img_dst);	1 // 複数色チャンネルの分割 2 Vector<Mat> img_bgr(3); 3 Vector<Mat> img_tmp(3); 4 split(img_src,img_tmp); 5 // 青→赤，緑→青，赤→緑に変更 6 img_bgr[0]=img_tmp[1]; 7 img_bgr[1]=img_tmp[2]; 8 img_bgr[2]=img_tmp[0]; 9 merge(img_bgr,img_dst);
54	8行目		cv2.cvtColor(img_src, cv2.COLOR_BGR2HSV);	cv2.cvtColor(img_src, cv2.COLOR_BGR2HSV)
87	19行目追加		**2**. 回転行列を組み合わせて，	**2**. 回転行列と平行移動を表すアフィン変換行列を組み合わせて，
93	33行目		14 double min, max;	14 double hist_min, hist_max;
	34行目		15 minMaxLoc(hist, &min, &max);	15 minMaxLoc(hist, &hist_min, &hist_max);
	41行目		(v/max)), Scalar(255, 255, 255));	(v/hist_max)), Scalar(255, 255, 255));
94	23行目		2 img_dst = np.zeros([100, 256]).	2 img_hst = np.zeros([100, 256]).
	24行目		3 rows, cols = img_dst.shape	3 rows, cols = img_hst.shape
	36行目		15 cv2.line(img_dst, (i, rows),	15 cv2.line(img_hst, (i, rows),
97	下から2行目		1 min = 100 2 max = 200 3 img_dst = cv2.equalizeHist(img_src)	1 img_dst = cv2.equalizeHist(img_src)
116	下から4行目		**4**. 真っ白，真っ黒な画像に対して誤差拡散ディザリングを行い，どのような出力画像が現れるか確認せよ.	**4**. 真っ白，単色グレー，真っ黒の画像に対して誤差拡散ディザリングを行い，どのような出力画像が現れるか確認せよ.
122		図7.5(c)	(11 × 11 ピクセル，$\sigma_1 = 1$，$\sigma_2 = 1$)	(11×11 ピクセル，$\sigma_1 = 50$，$\sigma_2 = 100$)
	6行目		2 bilateralFilter(img_src, img_dst, 11, 1, 1);	2 bilateralFilter(img_src, img_dst, 11, 50, 100);
123	14行目		2 img_dst = cv2.bilateralFilter(img_src, 11, 1, 1)	2 img_dst = cv2.bilateralFilter(img_src, 11, 50, 100)
133	2行目		1 Laplacian(img_src, img_tmp, CV_32F, 3); 2 convertScaleAbs(img_tmp, img_dst, 1, 0);	1 Mat img_tmp; 2 Laplacian(img_src, img_tmp, CV_32F, 3); 3 convertScaleAbs(img_tmp, img_dst, 1, 0);
136	11行目		7 filter2D(img_src, img_tmp, ddepth , op);	7 filter2D(img_src, img_tmp, CV_32F, op);
142	4行目		2 ret, img_dst = cv2.threshold(img_gray, thresh, 255, cv2.THRESH_BINARY)	2 ret, img_dst = cv2.threshold(img_src, thresh, 255, cv2.THRESH_BINARY)
147	2行目		膨張・収縮のプログラム例を以下に示す.	膨張・収縮のプログラム例を以下に示す．入力画像は二値画像とする.

148	4行目 〜 8行目		3 if((img_mask[y * width + x] == 255) 4 || (img_mask[y * width + (x - 1)] == 255) || (img_mask[y * width + (x + 1)] == 255) 5 || (img_mask[(y - 1) * width + x] == 255) || (img_mask[(y + 1) * width + x] == 255)) {img_dst[y * width + x] = 255;	3 if((img_src[y * width + x] == 255) 4 || (img_src[y * width + (x - 1)] == 255) || (img_src[y * width + (x + 1)] == 255) 5 || (img_src[(y - 1) * width + x] == 255) || (img_src[(y + 1) * width + x] == 255)) {img_dst[y * width + x] = 255;
148	17行目 〜 21行目		3 if((img_mask[y * width + x] == 0) 4 || (img_mask[y * width + (x - 1)] == 0) || (img_mask[y * width + (x + 1)] == 0) 5 || (img_mask[(y - 1) * width + x] == 0) || (img_mask[(y + 1) * width + x] == 0)) {img_dst[y * width + x] = 0;	3 if((img_src[y * width + x] == 0) 4 || (img_src[y * width + (x - 1)] == 0) || (img_src[y * width + (x + 1)] == 0) 5 || (img_src[(y - 1) * width + x] == 0) || (img_src[(y + 1) * width + x] == 0)) {img_dst[y * width + x] = 255;

150	19行目
	以降
			1 // オープニング
			2 morphologyEx(img_src, img_dst1, CV_MOP_OPEN,
			element8, Point(-1,-1), 1);
			3 // クロージング
			4 morphologyEx(img_dst1, img_dst2, CV_MOP_CLOSE,
			element8, Point(-1,-1), 1);
				1 Mat img_tmp;
				2 Mat element8=(Mat_<uchar>(3,3)
				<<1,1,1,1,1,1,1,1,1);// 8近傍
				3 // オープニング
				4 morphologyEx(img_src, img_tmp, CV_MOP_OPEN,
				element8, Point(-1,-1), 1);
				5 // クロージング
				6 morphologyEx(img_tmp, img_dst, CV_MOP_CLOSE,
				element8, Point(-1,-1), 1);

151	5行目		3 img_dst = cv2.morphologyEx(img_src, cv2. MORPH_OPEN, element8)	3 img_tmp = cv2.morphologyEx(img_src, cv2. MORPH_OPEN, element8)
	7行目		5 img_dst = cv2.morphologyEx(img_dst, cv2. MORPH_CLOSE, element8)	5 img_dst = cv2.morphologyEx(img_tmp, cv2. MORPH_CLOSE, element8)
153	17行目		外接長方形と縦横比を求めるプログラム例を以下に示す.	外接長方形と縦横比を求めるプログラム例を以下に示す. 入力画像は二値画像とする.

154	30行目
	〜
	34行目
			2 int x_min = img_bin.cols, x_max = 0;
			3 int y_min = img_bin.rows, y_max = 0;
			4 for(int y = 0; y < img_bin.rows; y++) {
			5 for(int x = 0; x < img_bin.cols; x++) {
			6 if(img_bin.data[y * img_bin.step + x] ==
			255) {
				2 int x_min = img_src.cols, x_max = 0;
				3 int y_min = img_src.rows, y_max = 0;
				4 for(int y = 0; y < img_src.rows; y++) {
				5 for(int x = 0; x < img_src.cols; x++) {
				6 if(img_src.data[y * img_src.step + x] ==
				255) {

155	2行目		1 rows, cols, ch = img_src.shape	1 rows, cols = img_src.shape
	3行目		2 print rows, cols, ch	2 print rows, cols
	12行目		11 if img_bin[y, x] == 255:	11 if img_src[y, x] == 255:
157	15行目		面積，周囲長，円形度の算出プログラム例を以下に示す.	面積，周囲長，円形度の算出プログラム例を以下に示す. 入力画像は二値画像とする.
162	2行目		重心と主軸角度を求めるプログラム例を以下に示す.	重心と主軸角度を求めるプログラム例を以下に示す. 入力画像は二値画像とする.
163	31行目		1 Moments m = moments(img_bin, true);	1 Moments m = moments(img_src, true);

177	2行目
	〜
	4行目
			1 Mat img_src1 // 入力画像1
			2 Mat img_src2 // 入力画像2
			3 Mat img_dst // 合成画像
				1 Mat img_src1; // 入力画像1
				2 Mat img_src2; // 入力画像2
				3 Mat img_dst; // 合成画像

181	下から 7行目		7 img_m = cv2.Threshold(img_df, 50, 255, cv2. THRESH_BINARY)	7 img_m = cv2.threshold(img_df, 50, 255, cv2. THRESH_BINARY)[1]
185	28行目 〜 29行目		9 img_df1b = cv2.threshold(img_df1,, 20, 255, cv2.THRESH_BINARY)[1] 10 img_df2b = cv2.threshold(img_df2,, 20, 255, cv2.THRESH_BINARY)[1]	9 img_df1b = cv2.threshold(img_df1, 20, 255, cv2.THRESH_BINARY)[1] 10 img_df2b = cv2.threshold(img_df2, 20, 255, cv2.THRESH_BINARY)[1]
186	2行目		**4**. 背景画像1枚と移動物体が移った3枚の入力画像を利用して，背景画像中に移動物体が連続して表示された出力画像を生成するプログラムを作成せよ.	**4**. 背景画像1枚と移動物体が写った3枚の入力画像を利用して，背景画像中に移動物体が連続して表示された出力画像を生成するプログラムを作成せよ.