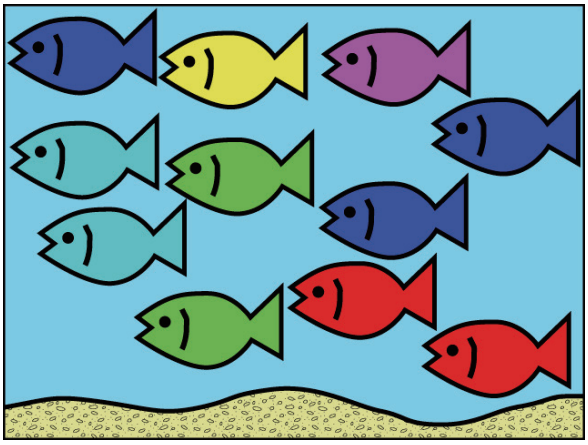
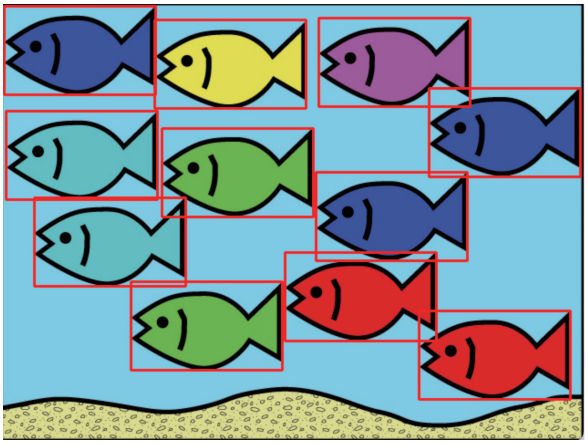
魚の検出



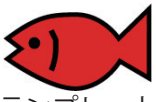
入力画像



出力画像

```
179 : 2匹
49 : 2匹
92 : 2匹
116 : 3匹
29 : 1匹
154 : 1匹
合計 : 11匹
```

出力



テンプレート画像

```cpp
//テンプレートマッチング
int TemplateMatching(const cv::Mat& src_img, const cv::Mat& template_img, cv::Mat& result_img,
    float TH) {
    cv::Mat compare_img;
    int count = 0;//魚の数カウント
    int a = 0;
    int X[100], Y[100];//座標が被っていないか確認用

    //類似度マップの作成
    compare_img = cv::Mat(cv::Size(src_img.cols - template_img.cols + 1, src_img.rows -
        template_img.rows + 1), CV_32F);
    cv::matchTemplate(src_img, template_img, compare_img, cv::TM_SQDIFF_NORMED);

    //検出
    for (int y = 0; y < compare_img.rows; y++) {
        for (int x = 0; x < compare_img.cols; x++) {
            float s = compare_img.at<float>(y, x);
            if (s <= TH) {
                int kensyutu = 0;//同じ魚を検出したら1
                for (int z = 0; z < a; z++) {//近くの座標だったら同じ魚だからカウントしない
                    if ((x - X[z] <= 10 && x - X[z] >= -10) && (y - Y[z] <= 10 && y - Y[z] >=
                        -10)) {
                        kensyutu = 1;
                        break;
                    }
                }
                if (kensyutu == 0) {//まだ検出されていない魚
                    //xy座標を記録
                    X[a] = x;
                    Y[a] = y;
                    a++;
                    count++;
                    cv::rectangle(result_img, cv::Point(x, y), cv::Point(x +
                        template_img.cols, y + template_img.rows), CV_RGB(255, 0, 0), 2);
                }
            }
        }
    }
    return count;//魚の数を返す
}
```

```cpp
int main (int argc, const char * argv[]) {
    cv::Mat src_img, hsv_img, template_red_img, template_yellow_img, template_green_img,
        template_lightblue_img, template_blue_img, template_purple_img, result_img;
    int count_y, count_r, count_g, count_l, count_b, count_p;

    //入力画像の読み込み
    src_img = cv::imread(TARGET_IMG_FILE, cv::IMREAD_COLOR);
    if (src_img.empty()) {
        fprintf(stderr, "File is not opened.\n");
        return (-1);
    }

    //テンプレート画像の読み込み
    template_red_img = cv::imread(RED_TEMPLATE_IMG_FILE, cv::IMREAD_COLOR);

    //各色の魚のテンプレートを作成
    cv::cvtColor(template_red_img, hsv_img, cv::COLOR_BGR2HSV);
    template_yellow_img = hsv_img - cv::Scalar(HUE_y, 0, 0);
    cv::cvtColor(template_yellow_img, template_yellow_img, cv::COLOR_HSV2BGR);
```

```cpp
    //結果画像の初期化
    result_img = src_img.clone();

    //テンプレートマッチング
    count_r = TemplateMatching(src_img, template_red_img, result_img, TH_r);
    count_g = TemplateMatching(src_img, template_green_img, result_img, TH_g);
    count_l = TemplateMatching(src_img, template_lightblue_img, result_img, TH_l);
    count_b = TemplateMatching(src_img, template_blue_img, result_img, TH_b);
    count_y = TemplateMatching(src_img, template_yellow_img, result_img, TH_y);
    count_p = TemplateMatching(src_img, template_purple_img, result_img, TH_p);

    //テンプレート画像の中央から色相を取得
    cv::cvtColor(template_red_img, hsv_img, cv::COLOR_BGR2HSV);
    int p_r = hsv_img.at<cv::Vec3b>(hsv_img.rows / 2, hsv_img.cols / 2)[0];
```

```cpp
    //出力
    cv::imshow("input", src_img); // 入力画像
    cv::imshow("result", result_img); // 結果画像

    //キー入力待ち
    cv::waitKey(0);

    return 0;
}
```

プログラム (一部)

同一形状、６色の魚が泳いでいる画像と、赤い魚の画像が入力される。このとき、魚の位置を検出し赤枠で囲んだ画像を出力し、同時にそれぞれの色相と魚の数を出力するプログラムを書いた。
テンプレート画像から６色の魚の画像を作り、作った各画像と出力画像をテンプレートマッチングし、魚を検出した。

Information

- 制作時間・・・・6 時間
- 制作時期・・・・2024 年 7 月
- 制作人数・・・・1 人
- 使用ソフト・・・Xcode
- 使用言語・・・・C++