

# それらしい物体をパッと見つける! リアルタイム色検出

森岡 澄夫

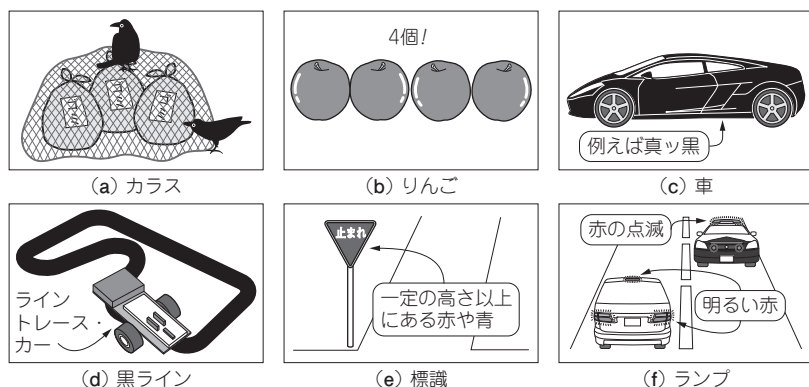


図1 特定の色を検出できるだけでいろいろなものが検出できるようになる

## ● 色情報を検出できると見つけられるものがたくさんある

画像処理の例として、画像中のどこに肌色の部分があるかを調べる処理(肌色検出)を組んでみます。ここで、多くの人が肌色検出と聞いて連想するのは、人や顔の検出かもしれません。肌色に限れば確かにそうなのですが、肌色検出の仕方を知れば、おのずと他の色の検出の仕方が分かります。つまり、本章で説明しようとしているのは人を見つける方法ではなく、色情報を使って物体を見つける方法です。図1に示すように実用、ホビーを問わず多くの局面に使えます。

## ● 色を検出できることの利点

肌色検出には以下のような用途があります。

- (1) 処理が軽いという利点を生かし、肌色をしているのが人しかいない状況や、それほど正確さが不必要な状況などでの簡易判定に使う[図2(a)]。
- (2) Raspberry Piのように本気で動画処理を行うにはCPU能力にさほど余裕がなく、いっばうで即応性が求められる場合、肌色部分の近傍だけを抜き出して顔検出にかけることで処理時間を短縮する[図2(b)]。
- (3) 色情報を使わない顔検出と併用し、検出された部分が肌色領域であるかを追加判定することで、誤検出の可能性を減らす[図2(c)]。

本稿では、この肌色検出を

- 方法1 OpenCVの関数のみで処理
- 方法2 自作関数とOpenCV関数のいいとこどりの2通りで試してみます。

## 判定を簡単にするための準備： 色空間の変換

### ● 赤/緑/青の原色を判定するのは割と簡単

入力されるのが普通のRGB画像であるとき、赤/緑/青など原色の物体を判定するのは簡単です。

例えば $R=255, G=0, B=0$ の全条件を満たす画素だけを残せば、赤い物体だけが画面に残ります(ただし現実には、255や0といった極端な値になることはないので、「 $R$ がある閾値以上、 $G$ と $B$ がある閾値以下」という条件にする)。

### ● 肌色検出はRGB空間でなくHSV空間のほうが簡単

しかし、肌色に対応する $R, G, B$ 値の範囲が一体何であるかは容易には分かりません。筆者も厳密に求めたことはないのですが、そもそも一つの連続した範囲に収まっておらず、複数の範囲を表す複雑な条件群になるかもしれません。

そこで、このような場合の常とう手段として、色空