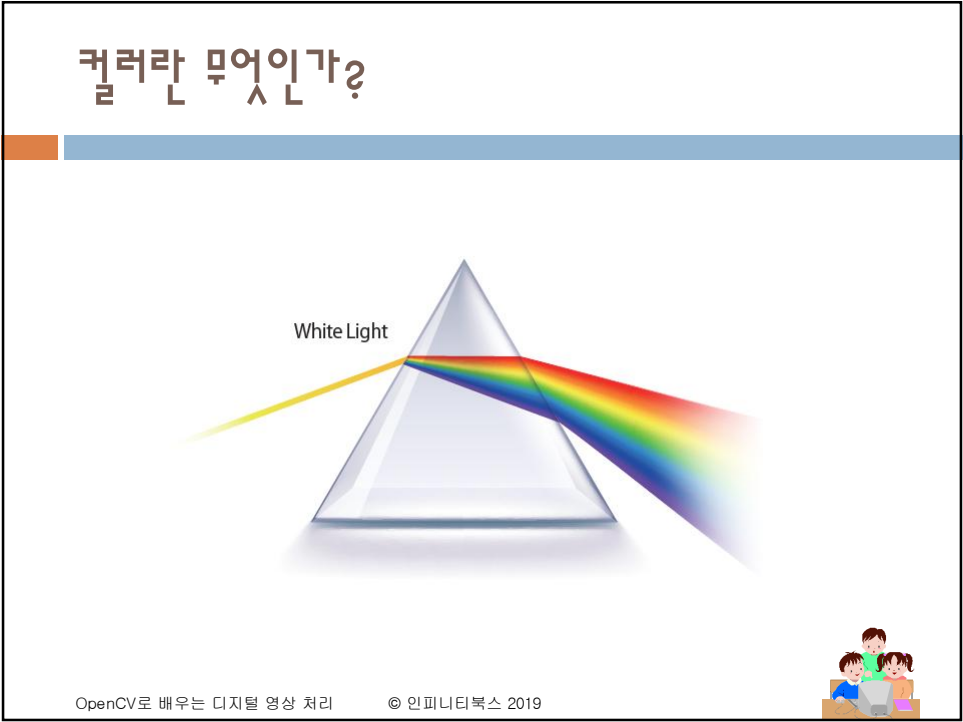
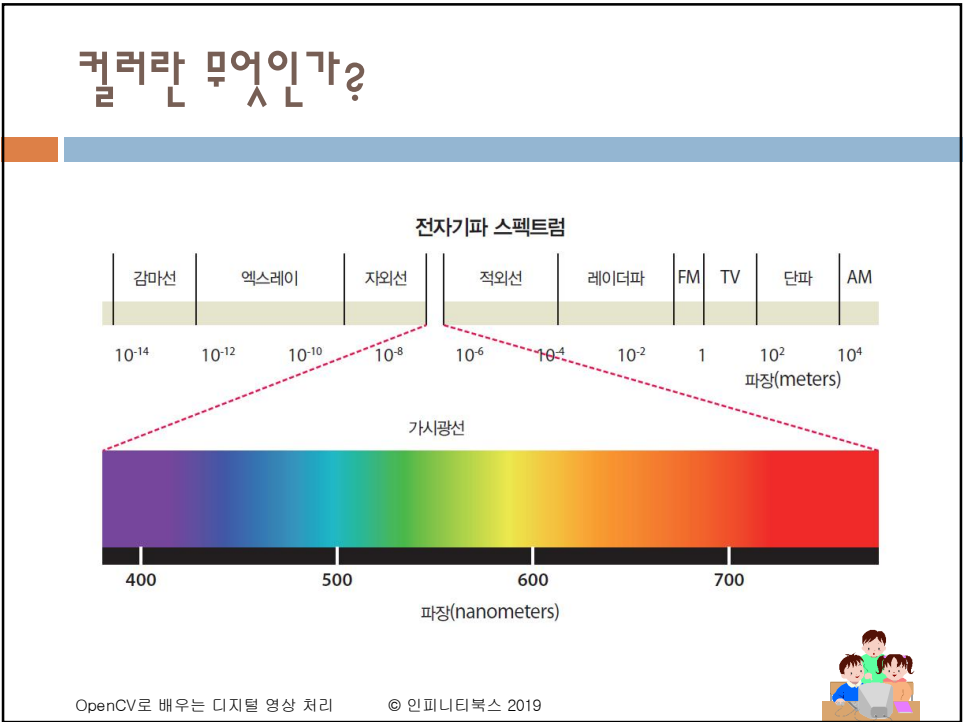


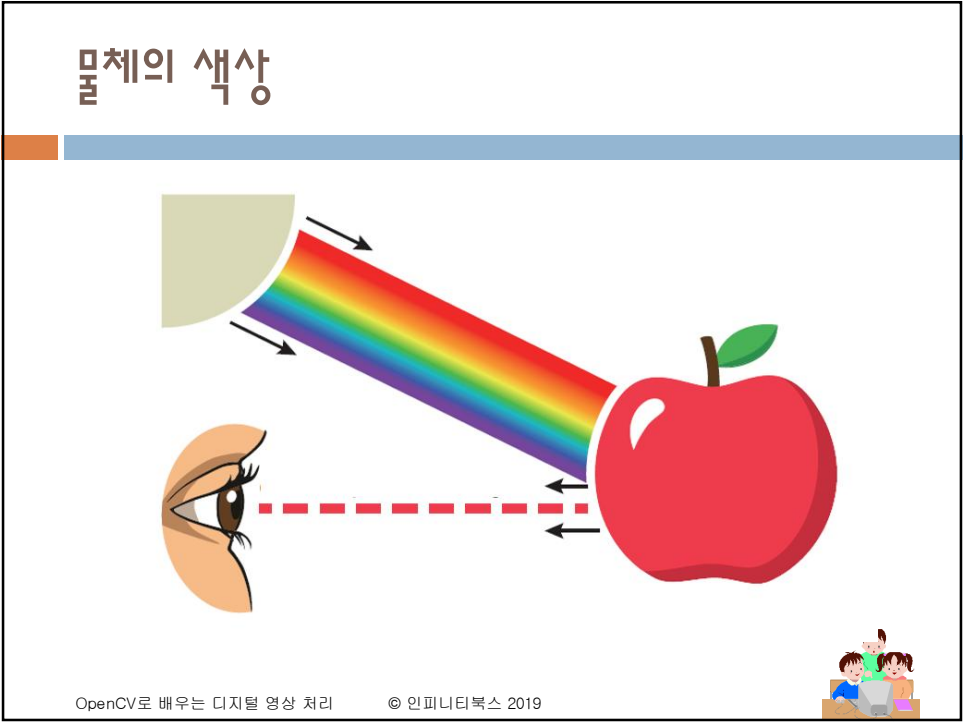
1



2



3



4

컬러와 인간의 시각

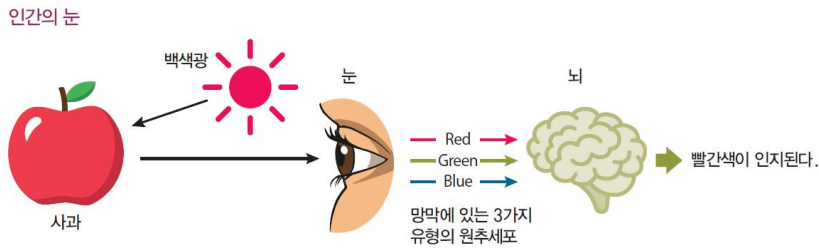
컬러	파장	주파수
Red	~ 700 – 635nm	~ 430 – 480THz
Orange	~ 635 – 590nm	~ 480 – 510THz
Yellow	~ 590 – 560nm	~ 510 – 540THz
Green	~ 560 – 520nm	~ 540 – 580THz
Cyan	~ 520 – 490nm	~ 580 – 610THz
Blue	~ 490 – 450nm	~ 610 – 670THz
Violet or Purple	~ 450 – 400nm	~ 670 – 750THz

OpenCV로 배우는 디지털 영상 처리 © 인피니티북스 2019



5

인간의 눈

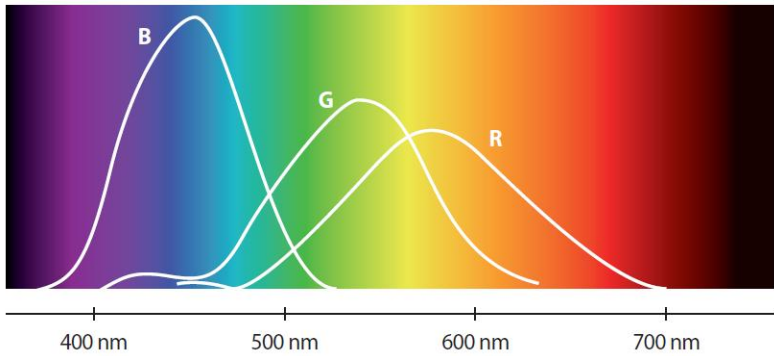


OpenCV로 배우는 디지털 영상 처리 © 인피니티북스 2019



6

3가지 원추 세포의 응답성

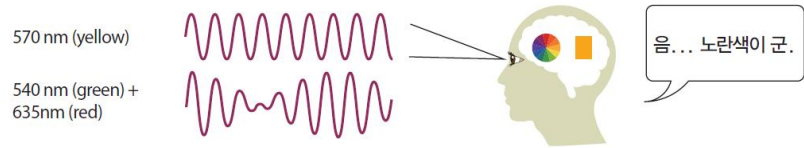


OpenCV로 배우는 디지털 영상 처리 © 인피니티북스 2019



7

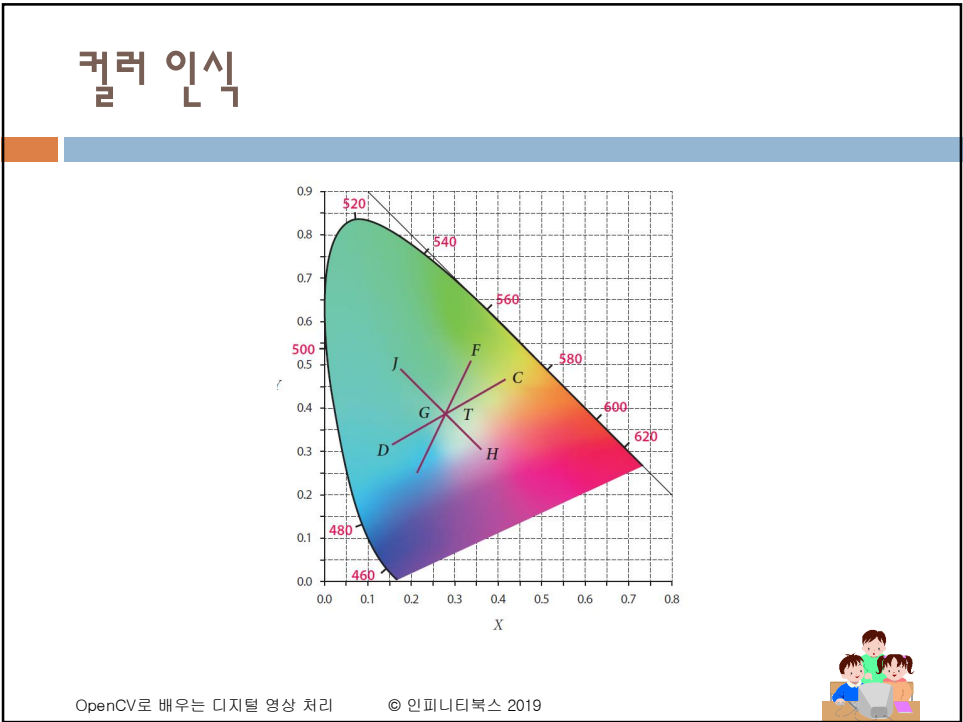
컬러 인식



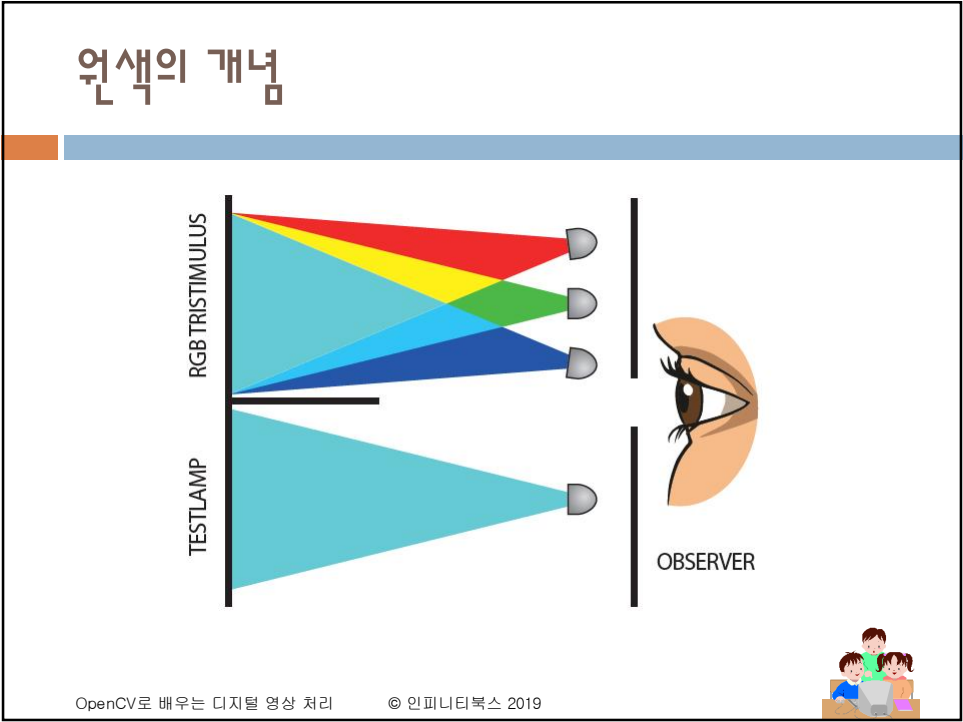
OpenCV로 배우는 디지털 영상 처리 © 인피니티북스 2019



8

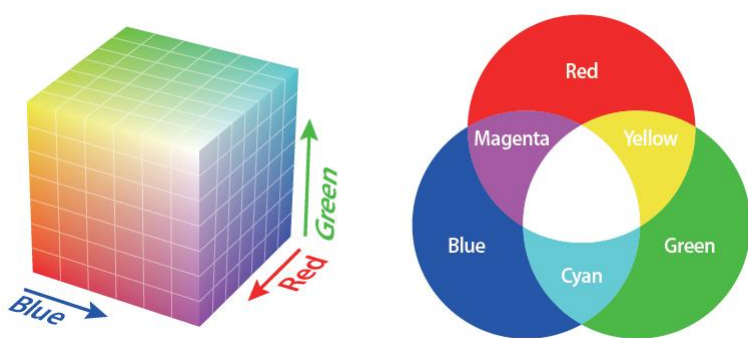


9




10

RGB 컬러 모델



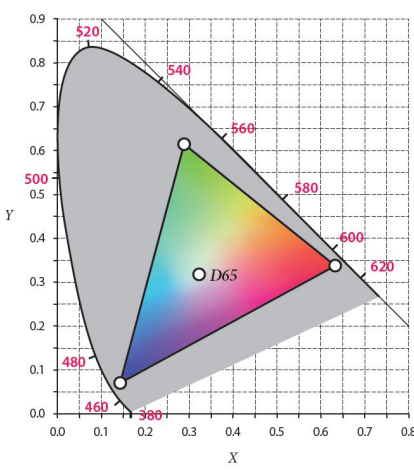
OpenCV로 배우는 디지털 영상 처리

© 인피니티북스 2019




11

RGB 컬러 모델



OpenCV로 배우는 디지털 영상 처리

© 인피니티북스 2019



12

CMY 컬러 모델

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

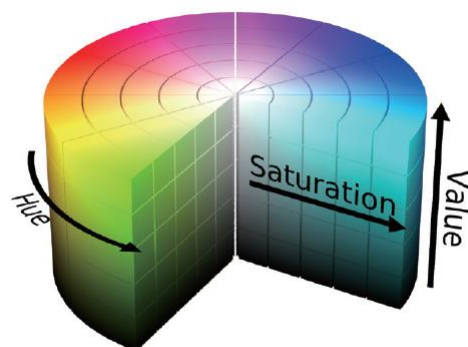
OpenCV로 배우는 디지털 영상 처리

© 인피니티북스 2019



13

HSV 컬러 모델



출처: 위키 백과

OpenCV로 배우는 디지털 영상 처리

© 인피니티북스 2019



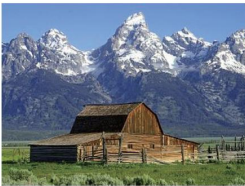
14

YIQ 컬러 모델

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.523 & 0.312 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$



YIQ 컬러 모델



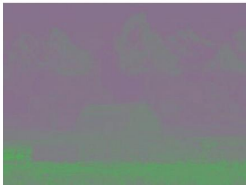
(원 사진)



(Y)



(I)



(Q)

By Tokachu at English Wikipedia, CC BY 2.5



색상 공간 변환

- BGR 영상에서 B, G, R을 분리하여 윈도우에 표시하려면 어떻게 해야 할까?
- -> `split()` 함수를 사용하면 된다.

OpenCV로 배우는 디지털 영상 처리

© 인피니티북스 2019



17

```
int main()
{
    Mat image;
    image = imread("d:/rcube.jpg", CV_LOAD_IMAGE_COLOR);

    Mat bgr[3];
    split(image, bgr);

    imshow("src", image);
    imshow("blue", bgr[0]);
    imshow("green", bgr[1]);
    imshow("red", bgr[2]);
    waitKey(0);
    return 0;
}
```

OpenCV로 배우는 디지털 영상 처리

© 인피니티북스 2019



18

실행 결과



OpenCV로 배우는 디지털 영상 처리

© 인피니티박스 2019



컬러 공간 변경

```
void cvtColor(InputArray src, OutputArray dst, int code)
```

매개 변수	설명
src	입력 영상
dst	출력 영상
code	색상 공간 변환 코드이다 • COLOR_BGR2HSV: BGR 컬러 모델을 HSV 컬러 모델로 변경한다. • COLOR_YUV2BGR: YUV 컬러 모델을 BGR 컬러 모델로 변경한다.

OpenCV로 배우는 디지털 영상 처리

© 인피니티박스 2019



```

int main()
{
    Mat image, hsv, dst;
    image = imread("d:/lenna.jpg", CV_LOAD_IMAGE_COLOR);
    cvtColor(image, hsv, CV_BGR2HSV);

    Mat array[3];
    split(hsv, array);
    imshow("src", image);
    imshow("hue", array[0]);
    imshow("saturation", array[1]);
    imshow("value", array[2]);

    waitKey(0);
    return 0;
}

```

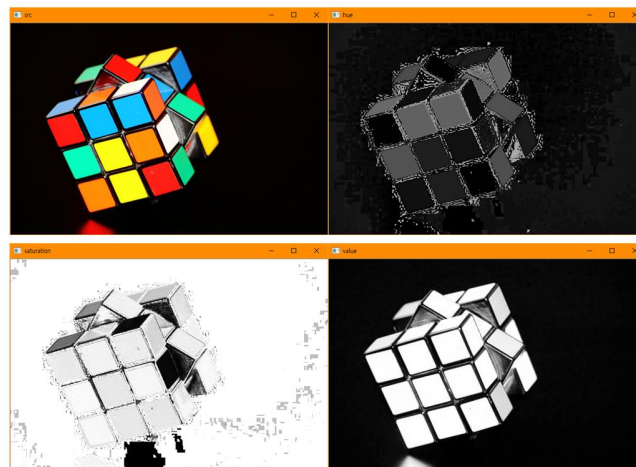
OpenCV로 배우는 디지털 영상 처리

© 인피니티북스 2019



21

실행 결과



OpenCV로 배우는 디지털 영상 처리

© 인피니티북스 2019






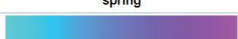









22

컬러맵 사용하기

- 영상의 컬러들을 다시 매핑하여서 영상의 분위기를 변경하는 연산
- $dst(x, y) \leftarrow T(src(x, y))$: 각 화소를 독립적으로 처리한다.
- $dst(x, y, c) \leftarrow LUT(src(x, y, c))$: 각 채널 당 하나의 참조 테이블을 사용한다.



applyColorMap()

COLORMAP_AUTUMN		COLORMAP_SPRING	
COLORMAP_BONE		COLORMAP_COOL	
COLORMAP_JET		COLORMAP_HSV	
COLORMAP_WINTER		COLORMAP_PINK	
COLORMAP_RAINBOW		COLORMAP_HOT	
COLORMAP_OCEAN		COLORMAP_PARULA	
COLORMAP_SUMMER			



```
#include "opencv2/opencv.hpp"
#include <iostream>
using namespace cv;
using namespace std;

int main()
{
    Mat img = imread("d:/lenna.jpg");
    Mat img_color;
    applyColorMap(img, img_color, COLORMAP_HOT);
    imshow("img_color", img_color);
    waitKey(0);
}
```

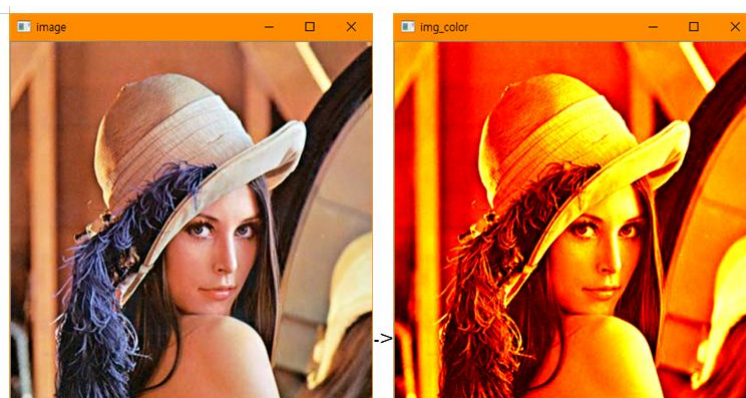
OpenCV로 배우는 디지털 영상 처리

© 인피니티북스 2019



25

실행 결과



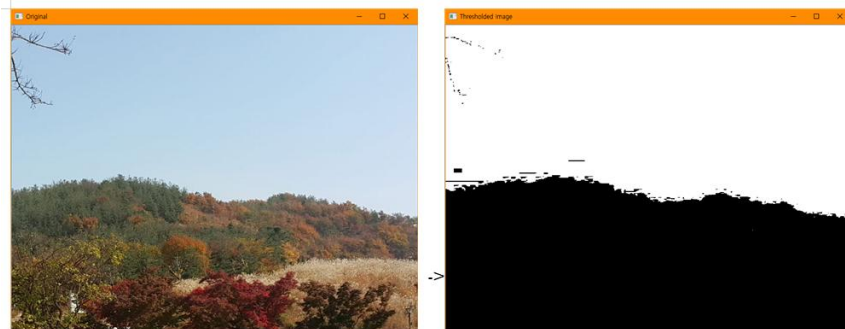
OpenCV로 배우는 디지털 영상 처리

© 인피니티북스 2019



26

컬러를 이용한 객체 분할



OpenCV로 배우는 디지털 영상 처리

© 인피니티북스 2019



27

```
int main()
{
    Mat img = imread("d:/image1.jpg", IMREAD_COLOR);
    if (img.empty()) { return -1; }

    Mat imgHSV;
    cvtColor(img, imgHSV, COLOR_BGR2HSV);

    Mat imgThresholded;
    inRange(imgHSV, Scalar(100, 0, 0),
            Scalar(120, 255, 255), imgThresholded);

    imshow("Thresholded Image", imgThresholded);
    imshow("Original", img);

    waitKey(0);
    return 0;
}
```

OpenCV로 배우는 디지털 영상 처리

© 인피니티북스 2019



28

크로마키 기법 구현하기

- 크로마키 합성(Chroma Key Composing)은 색조를 이용하여 2개의 영상 또는 비디오 스트림을 합성하는 기술이다.



OpenCV로 배우는 디지털 영상 처리

© 인피니티북스 2019



29

```
int main()
{
    Mat img = imread("d:/chroma.jpg", IMREAD_COLOR);
    Mat img2 = imread("d:/beach.png", IMREAD_COLOR);

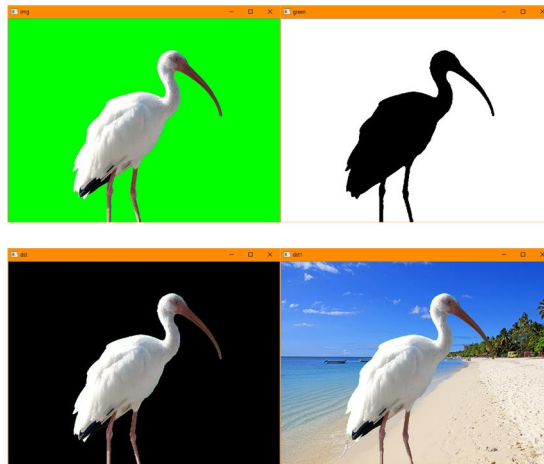
    Mat converted;
    cvtColor(img, converted, COLOR_BGR2HSV);
    Mat greenScreen = converted.clone();
    inRange(converted, Scalar(60-10, 100, 100), Scalar(60+10, 255, 255),
    greenScreen);

    Mat dst, dst1, inverted;
    bitwise_not(greenScreen, inverted);
    bitwise_and(img, img, dst, inverted);
    bitwise_or(dst, img2, dst1, greenScreen);
    bitwise_or(dst, dst1, dst1);

    imshow("img", img);
    imshow("green", greenScreen);
    imshow("dst", dst);
    imshow("dst1", dst1);
    waitKey(0);
    return 0;
}
```

30

컬러를 이용한 객체 분할



OpenCV로 배우는 디지털 영상 처리

© 인피니티북스 2019



31

컬러를 이용한 객체 추적

□ 비디오에서 색상을 이용하여 물체를 추적하는 방법을 살펴보자.

1. 비디오에서 프레임을 추출한다
2. **BGR** 색상 공간을 **HSV** 색상 공간으로 변환한다.
3. **HSV** 영상의 노랑색 구간을 이진화한다.
4. 노랑색 물체를 추출하여 어떤 작업을 한다.

OpenCV로 배우는 디지털 영상 처리

© 인피니티북스 2019



32


```

int main()
{
    VideoCapture cap("d:/tennis_ball.mp4");
    if (!cap.isOpened())
        return -1;
    for (;;)
    {
        Mat imgHSV;
        Mat frame;
        cap >> frame;
        cvtColor(frame, imgHSV, COLOR_BGR2HSV);

        Mat imgThresholded;
        inRange(imgHSV, Scalar(30, 10, 10), Scalar(38, 255, 255),
imgThresholded);

        imshow("frame", frame);
        imshow("dst", imgThresholded);
        if (waitKey(30) >= 0) break;
    }
    waitKey(0);
    return 0;
}

```

OpenCV로 배우는 디지털 영상 처리 © 인피니티북스 2019

33

실행 결과



OpenCV로 배우는 디지털 영상 처리 © 인피니티북스 2019

34

Q & A

