

표 4.1 이미지 입력 함수의 플래그

언어	플래그	설명
C#	ImreadModes.Unchanged	알파 채널을 포함해서 이미지 반환(알파 채널이 없을 경우 3채널로 반환)
Py	cv2.IMREAD_UNCHANGED	
C#	ImreadModes.Grayscale	단일 채널 그레이스케일 이미지로 변환
Py	cv2.IMREAD_GRAYSCALE	
C#	ImreadModes.Color	다중 채널 색상 이미지로 반환
Py	cv2.IMREAD_COLOR	
C#	ImreadModes.AnyDepth	정밀도가 존재할 경우 16/32비트 이미지로 반환(존재하지 않을 경우 8비트 이미지로 반환)
Py	cv2.IMREAD_ANYDEPTH	
C#	ImreadModes.AnyColor	채널이 존재할 경우 해당 채널 수로 반환(존재하지 않을 경우 3채널 이미지로 반환)
Py	cv2.IMREAD_ANYCOLOR	
C#	ImreadModes.ReducedGrayscale2	크기를 1/2로 줄인 후 그레이스케일 적용
Py	IMREAD_REDUCED_GRAYSCALE_2	
C#	ImreadModes.ReducedGrayscale4	크기를 1/4로 줄인 후 그레이스케일 적용
Py	cv2.IMREAD_REDUCED_GRAYSCALE_4	
C#	ImreadModes.ReducedGrayscale8	크기를 1/8로 줄인 후 그레이스케일 적용
Py	cv2.IMREAD_REDUCED_GRAYSCALE_8	
C#	ImreadModes.ReducedColor2	크기를 1/2로 줄인 후 다중 채널 색상 이미지로 반환
Py	cv2.IMREAD_REDUCED_COLOR_2	

언어	플래그	설명
C#	ImreadModes.IgnoreOrientation	EXIF의 방향 플래그에 따라 이미지를 회전하지 않음
Py	cv2.IMREAD_IGNORE_ORIENTATION	

예제 4.1 C# OpenCvSharp에서의 이미지 입력

```
using System;
using OpenCvSharp;

namespace Project
{
    class Program
    {
        static void Main(string[] args)
        {
            Mat src = Cv2.ImRead("OpenCV_Logo.png", ImreadModes.ReducedColor2);
            Console.WriteLine(src);
        }
    }
}
```

【출력 결과】

```
Mat [ 369*300*CV_8UC3, IsContinuous=True, IsSubmatrix=False, Ptr=0x12d4838, Data=0x60f58c0 ]
```

예제 4.2 Python OpenCV에서의 이미지 입력

```
import cv2

src = cv2.imread("OpenCV_Logo.png", cv2.IMREAD_GRAYSCALE)

print(src.ndim, src.shape, src.dtype)
```

[출력 결과]

2 (739, 600) uint8

예제 4.3 C# OpenCvSharp에서의 이미지 출력

```
using System;
using OpenCvSharp;

namespace Project
{
    class Program
    {
        static void Main(string[] args)
        {
            Mat src = Cv2.ImRead("OpenCV_Logo.png", ImreadModes.ReducedColor2);

            Cv2.NamedWindow("src", WindowMode.AutoSize);
            Cv2.SetWindowProperty("src", WindowProperty.Fullscreen, 0);
            Cv2.ImShow("src", src);
            Cv2.WaitKey(0);
            Cv2.DestroyWindow("src");
        }
    }
}
```

【출력 결과】



예제 4.4 Python OpenCV에서의 이미지 출력

```
import cv2

src = cv2.imread("OpenCV_Logo.png", cv2.IMREAD_GRAYSCALE)

cv2.namedWindow("src", flags=cv2.WINDOW_FREERATIO)
cv2.resizeWindow("src", 400, 200)
cv2.imshow("src", src)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

[출력 결과]

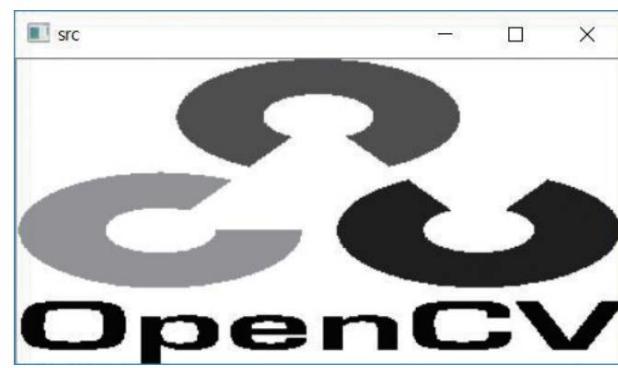


표 4.2 HighGUI 윈도우 함수

언어	함수	설명
C#	<pre>Cv2.NamedWindow(string winname, WindowMode flags)</pre>	<code>winname</code> 의 이름을 가지고 <code>flags</code> 로 설정된 윈도우를 생성
Py	<pre>cv2.namedWindow(winname, flags = None)</pre>	
C#	<pre>Cv2.MoveWindow(string winname, int x, int y)</pre>	<code>winname</code> 의 이름을 갖는 윈도우를 x, y 위치로 이동(좌측 상단 기준)
Py	<pre>cv2.moveWindow(winname, x, y)</pre>	

언어

함수

설명

```
Cv2.ResizeWindow(  
    string winname,  
    int width,  
    int height  
)
```

winname의 이름을 갖는 윈도우의 크기를 width, height로 설정

```
cv2.resizeWindow(  
    winname,  
    width,  
    height  
)
```

```
Cv2.SetWindowTitle(  
    string winname,  
    string title  
)
```

winname의 이름을 갖고 있는 윈도우의 이름을 title로 변경

```
cv2.setWindowTitle(  
    winname,  
    title  
)
```

Py

C#

Py

```
C# Cv2.SetWindowProperty(  
    string winname,  
    WindowProperty propId,  
    double propValue  
)
```

winname의 이름을 갖는 윈도우의 속성을 propId(prop_id)의 propValue(prop_value)로 변경

```
Py cv2.setWindowProperty(  
    winname,  
    prop_id,  
    prop_value  
)
```

```
C# Cv2.GetWindowProperty(  
    string winname,  
    WindowProperty propId  
)
```

winname의 이름을 갖는 윈도우의 속성 propId(prop_id)를 검색

```
Py cv2.getWindowProperty(  
    winname,  
    prop_id  
)
```

```
C# Cv2.StartWindowThread()
```

윈도우를 자동으로 업데이트하는 스레드를 실행

```
Py cv2.startWindowThread()
```

```
Cv2.DestroyWindow(
```

```
    string winname
```

```
)
```

winname의 이름을 갖는 윈도우를 제거

```
cv2.destroyWindow(
```

```
    winname
```

```
)
```

```
C# Cv2.DestroyAllWindows()
```

모든 윈도우를 제거

```
Py cv2.destroyAllWindows()
```

표 4.3 원도우의 플래그

언어	플래그	설명
C#	WindowMode.Normal	원도우 크기를 조절할 수 있으며, 최대화된 창을 이전 크기로 복원
Py	cv2.WINDOW_NORMAL	
C#	WindowMode.AutoSize	원도우 크기를 조절할 수 없으며, 이미지의 크기와 동일하게 표시
Py	cv2.WINDOW_AUTOSIZE	
C#	WindowMode.FullScreen	원도우를 최대화
Py	cv2.WINDOW_FULLSCREEN	
C#	WindowMode.KeepRatio	이미지 비율을 최대한 유지
Py	cv2.WINDOW_KEEP_RATIO	
C#	WindowMode.FreeRatio	비율의 제한이 없는 경우 이미지를 최대한 확장
Py	cv2.WINDOW_FREERATIO	
C#	WindowMode.OpenGL	OpenGL ⁵ 을 지원하는 원도우
Py	cv2.WINDOW_OPENGL	

표 4.4 윈도우의 속성

언어	속성	세부 속성	설명
C#	WindowProperty.Fullscreen	0	최대화
Py	cv2.WND_PROP_FULLSCREEN	1	전체 화면으로 표시
C#	WindowProperty.AutoSize	0	윈도우의 크기를 출력된 이미지 크기로 조절
Py	cv2.WND_PROP_AUTOSIZE	1	윈도우 크기에 맞게 이미지 크기를 조절
C#	WindowProperty.AspectRatio	0	이미지 비율을 최대한 유지
Py	cv2.WND_PROP_ASPECT_RATIO	1	비율의 제한이 없는 경우 이미지를 최대한 확장
C#	WindowProperty.OpenGL	0	OpenGL을 지원하는 윈도우
Py	cv2.WND_PROP_OPENGL		

예제 4.5 C# OpenCvSharp에서의 동영상 출력

```
using System;
using OpenCvSharp;

namespace Project
{
    class Program
    {
        static void Main(string[] args)
        {
            VideoCapture capture = new VideoCapture("Star.mp4");
            Mat frame = new Mat();

            while(true)
            {
                if (capture.PosFrames == capture.FrameCount) capture.Open("star.mp4");

                capture.Read(frame);
                Cv2.ImShow("VideoFrame", frame);

                if (Cv2.WaitKey(33) == 'q') break;
            }

            capture.Release();
            Cv2.DestroyAllWindows();
        }
    }
}
```

예제 4.6 Python OpenCV에서의 동영상 출력

```
import cv2

capture = cv2.VideoCapture("Star.mp4")

while True:
    ret, frame = capture.read()

    if(capture.get(cv2.CAP_PROP_POS_FRAMES) == capture.get(cv2.CAP_PROP_FRAME_COUNT)):
        capture.open("Star.mp4")

    cv2.imshow("VideoFrame", frame)
    if cv2.waitKey(33) == ord('q'): break

capture.release()
cv2.destroyAllWindows()
```

$$\text{FPS} = \frac{1000}{\text{Interval}}$$

$$\text{FPS} = \frac{1000}{\text{Interval}} = \frac{1000}{33} = 30.303030\dots \approx 30$$

표 4.5 동영상 출력 클래스의 메서드

언어	메서드	설명
C#	capture.IsOpened()	동영상 파일 열기의 성공 여부 확인
Py	capture.isOpened()	
	capture.Read()	
C#	Mat mat) 	동영상 파일에서 프레임을 읽음
Py	capture.read()	
	capture.Open()	
C#	string fileName) 	동영상 파일을 읽음
Py	capture.open() fileName)	
	capture.Set(CaptureProperty propertyId, double value) 	동영상 파일 속성 설정. propertyId(propid)의 속성을 value 값 으로 할당
C#	capture.set(propid, value) 	
Py		

```
C# capture.Get(
```

```
    CaptureProperty propertyId
```

```
)
```

동영상 파일의 속성을 반환. propertyId(propid)의 속성에 할당된
값을 반환

```
capture.get(
```

```
    propid
```

```
)
```

```
C# capture.Release()
```

동영상 파일을 닫고 메모리를 해제

```
Py capture.release()
```

표 4.6 동영상 출력 클래스 속성

언어	속성	설명
C#	CaptureProperty.PosMsec	동영상의 현재 위치(밀리초) 또는 타임스탬프
Py	cv2.CAP_PROP_POS_MSEC	
C#	CaptureProperty.PosFrames	동영상의 현재 프레임
Py	cv2.CAP_PROP_POS_FRAMES	
C#	CaptureProperty.PosAviRatio	동영상 내의 상대적 위치(0.0~1.0)
Py	CAP_PROP_POS_AVI_RATIO	
C#	CaptureProperty.FrameWidth	동영상 프레임 너비
Py	cv2.CAP_PROP_FRAME_WIDTH	
C#	CaptureProperty.FrameHeight	동영상 프레임 높이
Py	cv2.CAP_PROP_FRAME_HEIGHT	
C#	CaptureProperty.Fps	동영상의 프레임 속도
Py	cv2.CAP_PROP_FPS	
C#	CaptureProperty.FourCC	동영상의 코덱
Py	cv2.CAP_PROP_FOURCC	
C#	CaptureProperty.FrameCount	동영상의 총 프레임 수
Py	cv2.CAP_PROP_FRAME_COUNT	
C#	CaptureProperty.Mode	캡처 모드를 나타내는 백엔드 관련 값
Py	cv2.CAP_PROP_MODE	

C#	CaptureProperty.Brightness	카메라의 밝기 설정(지원되는 카메라에 한함)
Py	cv2.CAP_PROP_BRIGHTNESS	
C#	CaptureProperty.Contrast	카메라의 대비 설정(지원되는 카메라에 한함)
Py	cv2.CAP_PROP_CONTRAST	
C#	CaptureProperty.Saturation	카메라의 채도 설정(지원되는 카메라에 한함)
Py	cv2.CAP_PROP_SATURATION	
C#	CaptureProperty.Hue	카메라의 색상 설정(지원되는 카메라에 한함)
Py	cv2.CAP_PROP_HUE	
C#	CaptureProperty.Gain	카메라의 개인 설정(지원되는 카메라에 한함)
Py	cv2.CAP_PROP_GAIN	
C#	CaptureProperty.Exposure	카메라의 노출 설정(지원되는 카메라에 한함)
Py	cv2.CAP_PROP_EXPOSURE	

예제 4.7 C# OpenCvSharp에서의 카메라 출력

```
using System;
using OpenCvSharp;

namespace Project
{
    class Program
    {
        static void Main(string[] args)
        {
            VideoCapture capture = new VideoCapture(0);
            Mat frame = new Mat();
            capture.Set(CaptureProperty.FrameWidth, 640);
            capture.Set(CaptureProperty.FrameHeight, 480);

            while(true)
            {
                if (capture.IsOpened() == true)
                {
                    capture.Read(frame);
                    Cv2.ImShow("VideoFrame", frame);
                    if (Cv2.WaitKey(33) == 'q') break;
                }
            }

            capture.Release();
            Cv2.DestroyAllWindows();
        }
    }
}
```

예제 4.8 Python OpenCV에서의 카메라 출력

```
import cv2

capture = cv2.VideoCapture(0)
capture.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
capture.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)

while True:
    ret, frame = capture.read()
    if ret == True:
        cv2.imshow("VideoFrame", frame)
        if cv2.waitKey(33) == ord('q'): break
    else:
        break

capture.release()
cv2.destroyAllWindows()
```

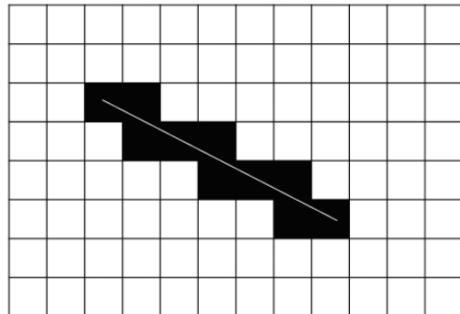
표 4.7 동영상 출력 클래스 메서드

언어	메서드	설명
C#	capture.Open(int index)	카메라 장치를 읽음
Py	capture.open(index)	
C#	bool = capture.Grab()	동영상 파일에서 프레임을 읽어 내부 버퍼에 저장
Py	bool = capture.grab()	
C#	capture.Retrieve(Mat mat, CameraChannels flag)	멀티 헤드 카메라의 프레임을 디코딩한 후 반환
Py	capture.retrieve(ndarray, flag)	

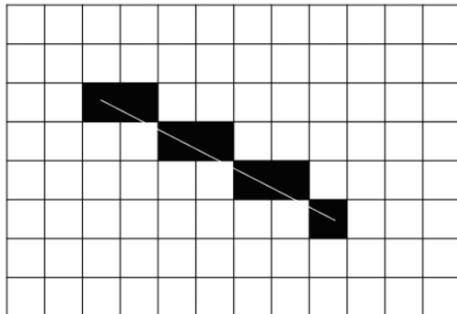


그림 4.1 도형 그리기

(a) 브레젠험 4연결 방식



(b) 브레젠험 8연결 방식



(c) 안티 에일리어싱 방식

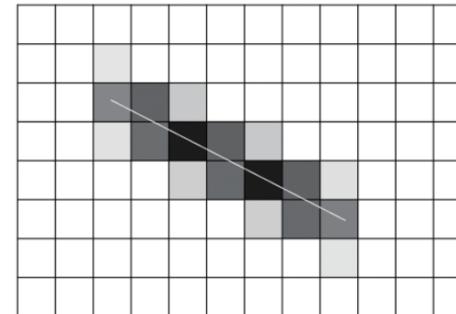
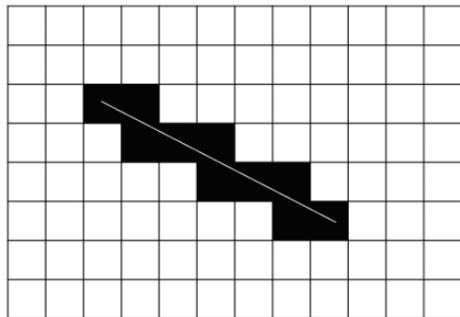
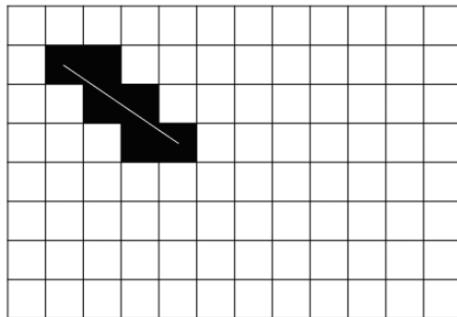


그림 4.2 선형 타입의 종류

(a) 0-shift



(b) 1-shift



(c) 2-shift

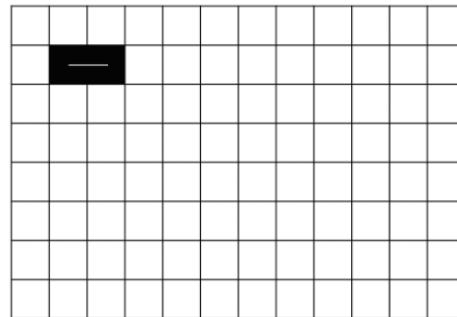


그림 4.3 비트 시프트 연산

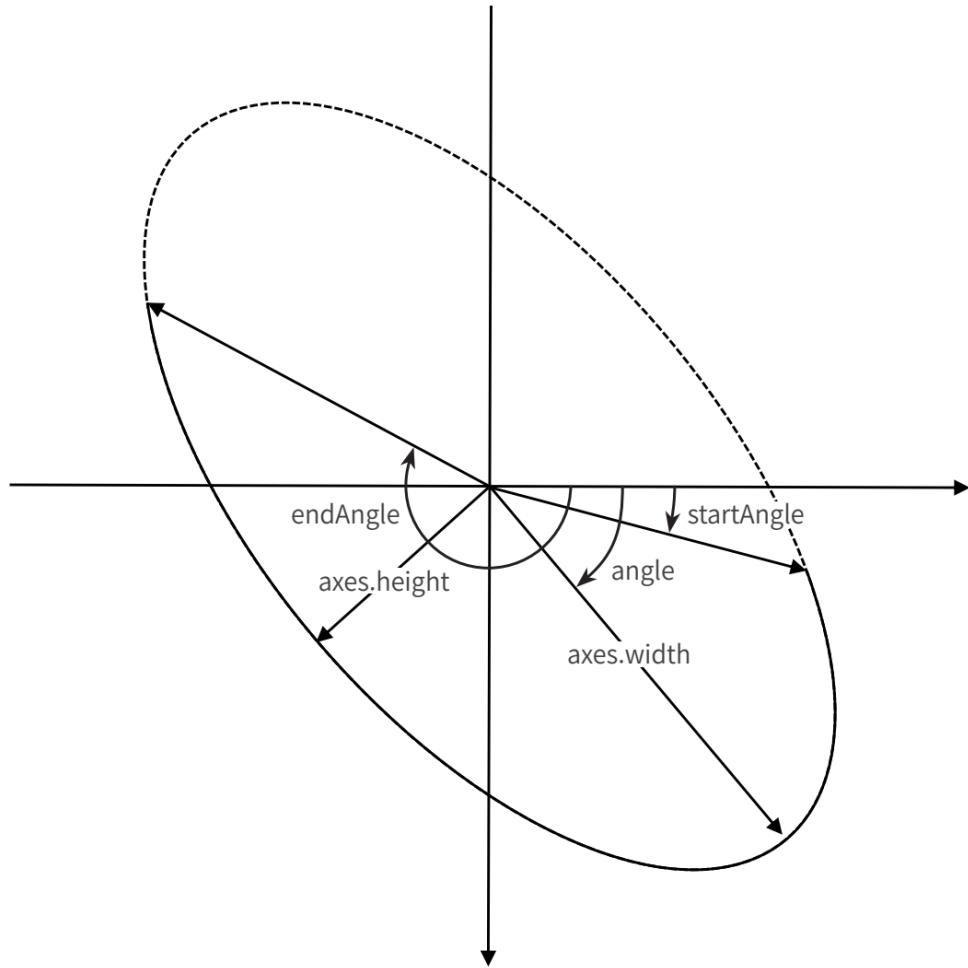


그림 4.4 호 그리기 함수에 사용되는 매개변수의 의미

표 4.8 OpenCV에서 사용하는 글꼴

언어	속성	설명
C#	HersheyFonts.HersheySimplex	보통 크기의 산세리프 글꼴
Py	cv2.FONT_HERSHEY_SIMPLEX	
C#	HersheyFonts.HersheyPlain	작은 크기의 산세리프 글꼴
Py	cv2.FONT_HERSHEY_PLAIN	
C#	HersheyFonts.HersheyDuplex	정교한 보통 크기의 산세리프 글꼴
Py	cv2.FONT_HERSHEY_DUPLEX	
C#	HersheyFonts.HersheyComplex	보통 크기의 세리프 글꼴
Py	cv2.FONT_HERSHEY_COMPLEX	
C#	HersheyFonts.HersheyTriplex	정교한 보통 크기의 세리프 글꼴
C#	cv2.FONT_HERSHEY_TRIPLEX	
C#	HersheyFonts.HersheyComplexSmall	작은 크기의 세리프 글꼴
Py	cv2.FONT_HERSHEY_COMPLEX_SMALL	
C#	HersheyFonts.HersheyScriptSimplex	필기체 스타일 글꼴
Py	cv2.FONT_HERSHEY_SCRIPT_SIMPLEX	
C#	HersheyFonts.HersheyScriptComplex	정교한 필기체 스타일 글꼴
Py	cv2.FONT_HERSHEY_SCRIPT_COMPLEX	
C#	HersheyFonts.Italic	기울임꼴
Py	cv2.FONT_ITALIC	

예제 4.9 C# OpenCvSharp의 그리기 함수 활용

```
    new Point(200, 600)
};

List<Point> pt2 = new List<Point>()
{
    new Point(400, 500),
    new Point(500, 500),
    new Point(600, 700),
    new Point(500, 650)

};

pts1.Add(pt1);
pts1.Add(pt2);
Cv2.Polylines(img, pts1, true, new Scalar(0, 255, 255), 2);

Point[] pt3 = new Point[] {
    new Point(700, 500),
    new Point(800, 500),
    new Point(700, 600)
};

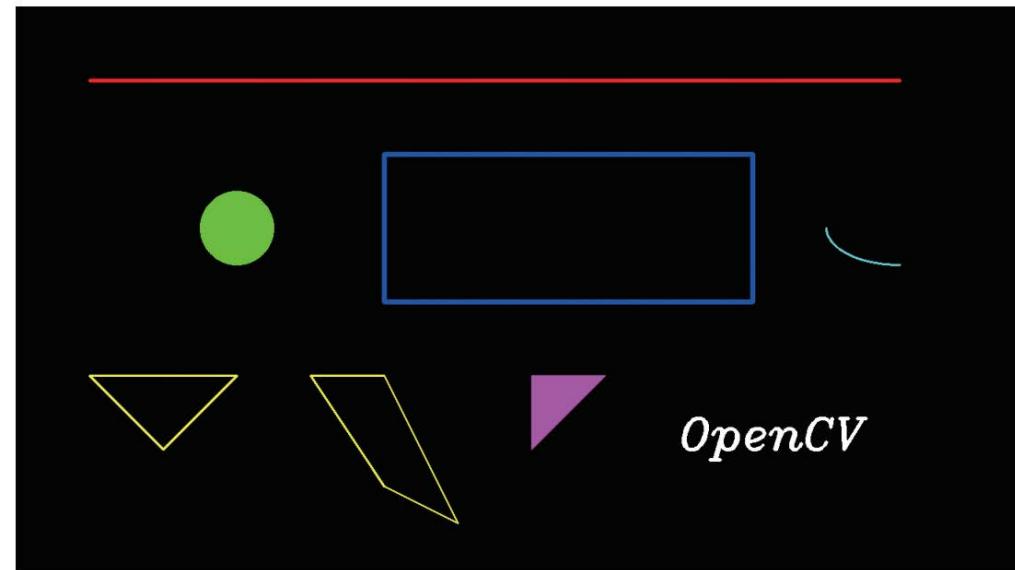
Point[][] pts2 = new Point[][] { pt3 };

Cv2.FillPoly(img, pts2, new Scalar(255, 0, 255), LineTypes.AntiAlias);

Cv2.PutText(img, "OpenCV", new Point(900, 600), HersheyFonts.HersheyComplex |
HersheyFonts.Italic, 2.0, new Scalar(255, 255, 255), 3);
```

```
Cv2.ImShow("img", img);
Cv2.WaitKey(0);
Cv2.DestroyAllWindows();
}
}
}
```

【출력 결과】



예제 4.10 Python OpenCV의 그리기 함수 활용

```
import numpy as np
import cv2

img = np.zeros((768, 1366, 3), dtype = np.uint8)

cv2.line(img, (100, 100), (1200, 100), (0, 0, 255), 3, cv2.LINE_AA)
cv2.circle(img, (300, 300), 50, (0, 255, 0), cv2.FILLED, cv2.LINE_4)
cv2.rectangle(img, (500, 200), (1000, 400), (255, 0, 0), 5, cv2.LINE_8)
cv2.ellipse(img, (1200, 300), (100, 50), 0, 90, 180, (255, 255, 0), 2)

pts1 = np.array([[[100, 500], [300, 500], [200, 600]], [[400, 500], [500, 500], [600, 700]]])
pts2 = np.array([[700, 500], [800, 500], [700, 600]])
cv2.polyline(img, pts1, True, (0, 255, 255), 2)
cv2.fillPoly(img, [pts2], (255, 0, 255), cv2.LINE_AA)

cv2.putText(img, "OpenCV", (900, 600), cv2.FONT_HERSHEY_COMPLEX | cv2.FONT_ITALIC, 2, (255, 255, 255), 3)

cv2.imshow("img", img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

[출력 결과]

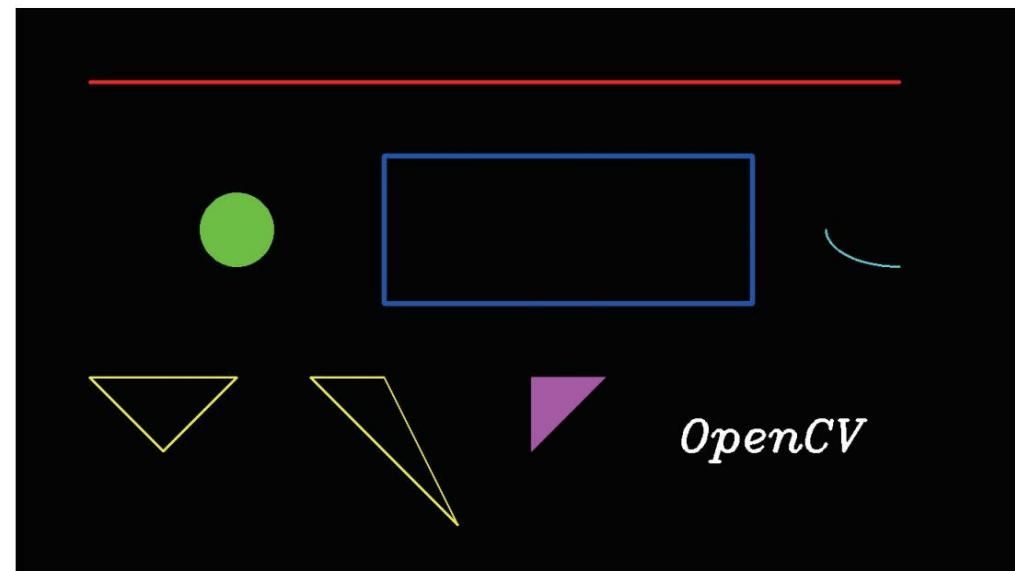


표 4.9 인코딩 매개변수

언어	속성	설명
C#	ImwriteFlags.JpegQuality	JPEG 화질(0~100)
Py	cv2.IMWRITE_JPEG_QUALITY	
C#	ImwriteFlags.JpegProgressive	점차 선명해짐 적용(0, 1)
Py	cv2.IMWRITE_JPEG_PROGRESSIVE	
C#	ImwriteFlags.JpegOptimize	최적화 적용(0, 1)
Py	cv2.IMWRITE_JPEG_OPTIMIZE	
C#	ImwriteFlags.JpegRstInterval	마커의 간격 설정(0~65535)
Py	cv2.IMWRITE_JPEG_RST_INTERVAL	
C#	ImwriteFlags.JpegLumaQuality	루마 품질 적용(0~100)
C#	cv2.IMWRITE_JPEG_LUMA_QUALITY	
C#	ImwriteFlags.JpegChromaQuality	크로마 품질 적용(0~100)
Py	cv2.IMWRITE_JPEG_CHROMA_QUALITY	
C#	ImwriteFlags.PngCompression	PNG 압축(0~100)
Py	cv2.IMWRITE_PNG_COMPRESSION	
C#	ImwriteFlags.PngBilevel	바이너리 포맷 사용(0, 1)
Py	cv2.IMWRITE_PNG_BILEVEL	
C#	ImwriteFlags.PxmBinary	PPM, PGM, PBM 파일을 바이너리 포맷 사용(0, 1)
Py	cv2.IMWRITE_PXM_BINARY	

C#	ImwriteFlags.WebPQuality	WebP 적용(0~100)
Py	cv2.IMWRITE_WEBP_QUALITY	
C#	ImwriteFlags.TiffResUnit	TIFF 사용(DPI 값)
Py	cv2.IMWRITE_TIFF_RESUNIT	
C#	ImwriteFlags.TiffXdpi	TIFF 포맷의 X 방향 DPI(DPI 값)
Py	cv2.IMWRITE_TIFF_XDPI	
C#	ImwriteFlags.TiffYDpi	TIFF 포맷의 Y 방향 DPI(DPI 값)
Py	cv2.IMWRITE_TIFF_YDPI	

예제 4.11 C# OpenCvSharp에서의 이미지 저장

```
ImageEncodingParam[] prms = new ImageEncodingParam[] {  
    new ImageEncodingParam(ImwriteFlags.JpegQuality, 100),  
    new ImageEncodingParam(ImwriteFlags.JpegProgressive, 1)  
};  
save = Cv2.ImWrite("CV.jpeg", img, prms);  
Console.WriteLine(save);
```

[출력 결과]

True

예제 4.12 Python OpenCV에서의 이미지 저장

```
save = cv2.imwrite("CV.jpeg", img, (cv2.IMWRITE_JPEG_QUALITY, 100, cv2.IMWRITE_JPEG_PROGRESSIVE, 1))  
print(save)
```

[출력 결과]

True

표 4.10 FourCC 디지털 포맷 코드

코덱	int 형식	코덱	int 형식
Prompt	-1	DIB	541215044
IV32	842225225	H261	825635400
IV41	825513545	H263	859189832
IV50	808801865	H264	875967048
IYUB	1448433993	MSVC	1129730893
PIM1	827148624	MJPEG	1196444237
CVID	1145656899	MPG4	877088845
XVID	1145656920	MP42	842289229
DIVX	1482049860	MP43	859066445

예제 4.13 C# OpenCvSharp에서의 동영상 저장

```
using System;
using OpenCvSharp;

namespace Project
{
    class Program
    {
        static void Main(string[] args)
        {
            VideoCapture capture = new VideoCapture("Star.mp4");
            Mat frame = new Mat(new Size(capture.FrameWidth, capture.FrameHeight), MatType.CV_8UC3);
            VideoWriter videoWriter = new VideoWriter();
            bool isWrite = false;

            while (true)
            {
                if (capture.PosFrames == capture.FrameCount) capture.Open("Star.mp4");

                capture.Read(frame);
                Cv2.ImShow("VideoFrame", frame);

                int key = Cv2.WaitKey(33);
                if (key == 4)
                {
                    videoWriter.Open("Video.avi", FourCC.XVID, 30, frame.Size(), true);
                }
            }
        }
    }
}
```

```
    isWrite = true;

}

else if (key == 24)
{
    videoWriter.Release();
    isWrite = false;
}

else if (key == 'q') break;

if (isWrite == true) videoWriter.Write(frame);
}

videoWriter.Release();
capture.Release();
Cv2.DestroyAllWindows();

}

}
```

예제 4.14 Python OpenCV에서의 동영상 저장

```
import cv2

capture = cv2.VideoCapture("Star.mp4")
width = int(capture.get(cv2.CAP_PROP_FRAME_WIDTH))
height = int(capture.get(cv2.CAP_PROP_FRAME_HEIGHT))
videoWriter = cv2.VideoWriter()
isWrite = False

while True:
    ret, frame = capture.read()

    if(capture.get(cv2.CAP_PROP_POS_FRAMES) == capture.get(cv2.CAP_PROP_FRAME_COUNT)):
        capture.open("Star.mp4")

    cv2.imshow("VideoFrame", frame)
    key = cv2.waitKey(33)

    if key == 4:
        fourcc = cv2.VideoWriter_fourcc(*'XVID')
        videoWriter.open("Video.avi", fourcc, 30, (width, height), True)
        isWrite = True

    elif key == 24:
        videoWriter.release()
```

```
isWrite = False

elif key == ord('q'): break

if isWrite == True:
    videoWriter.write(frame)

videoWriter.release()
capture.release()
cv2.destroyAllWindows()
```

표 4.11 동영상 저장과 관련된 함수

언어	속성	설명
C#	videoWriter.IsOpened()	동영상 저장의 성공 여부 확인
Py	videoWriter.isOpened()	
C#	videoWriter.Write(Mat mat)	동영상 파일에 프레임을 저장
Py	videoWriter.write(ndarray)	
C#	videoWriter.Open(string fileName, FourCC fourcc, double fps, Size frameSize, bool isColor = true)	동영상 저장 구조 생성
Py	videoWriter.open(fileName, fourcc, fps, frameSize, isColor = True)	
C#	videoWriter.Release()	동영상 저장 구조 메모리 해제
Py	videoWriter.release()	

예제 5.1 C# OpenCvSharp에서의 색상 공간 변환

```
using System;
using OpenCvSharp;

namespace Project
{
    class Program
    {
        static void Main(string[] args)
        {
            Mat src = Cv2.ImRead("crow.jpg");
            Mat dst = new Mat(src.Size(), MatType.CV_8UC1);

            Cv2.CvtColor(src, dst, ColorConversionCodes.BGR2GRAY);

            Cv2.ImShow("dst", dst);
            Cv2.WaitKey(0);
            Cv2.DestroyAllWindows();
        }
    }
}
```

[출력 결과]



예제 5.2 Python OpenCV에서의 색상 공간 변환

```
import cv2

src = cv2.imread("crow.jpg")
dst = cv2.cvtColor(src, cv2.COLOR_BGR2HSV)

cv2.imshow("dst", dst)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

[출력 결과]



표 5.1 색상 변환 코드

언어	속성	설명
C#	ColorConversionCodes.BGR2RGB	
	ColorConversionCodes.BGRA2RGBA	
	ColorConversionCodes.RGB2BGR	
	ColorConversionCodes.RGBA2BGRA	BGR(A)과 RGB(A) 색상 공간 상호 변환
Py	cv2.COLOR_BGR2RGB	
	cv2.COLOR_BGRA2RGBA	
	cv2.COLOR_RGB2BGR	
	cv2.COLOR_RGBA2BGRA	
C#	ColorConversionCodes.BGR2BGRA	
	ColorConversionCodes.BGR2RGBA	
	ColorConversionCodes.RGB2BGR	
	ColorConversionCodes.RGB2RGB	BGR0 나 RGB 색상 공간에 알파 채널 추가(색상 공간 변환 포함)
Py	cv2.COLOR_BGR2BGRA	
	cv2.COLOR_BGR2RGBA	
	cv2.COLOR_RGB2BGR	
	cv2.COLOR_RGB2RGBA	
C#	ColorConversionCodes.BGRA2BGR	
	ColorConversionCodes.BGRA2RGB	
	ColorConversionCodes.RGBA2BGR	
	ColorConversionCodes.RGBA2RGB	BGRA0 나 RGBA 색상 공간에 알파 채널 제거(색상 공간 변환 포함)
Py	cv2.COLOR_BGRA2BGR	
	cv2.COLOR_BGRA2RGB	
	cv2.COLOR_RGBA2BGR	
	cv2.COLOR_RGBA2RGB	

C#	ColorConversionCodes.BGR2GRAY ColorConversionCodes.BGRA2GRAY ColorConversionCodes.RGB2GRAY ColorConversionCodes.RGBA2GRAY	BGR(A)나 RGB(A) 색상 공간을 그레이스케일로 변환
Py	cv2.COLOR_BGR2GRAY cv2.COLOR_BGRA2GRAY cv2.COLOR_RGB2GRAY cv2.COLOR_RGBA2GRAY	
C#	ColorConversionCodes.GRAY2BGR ColorConversionCodes.GRAY2BGRA ColorConversionCodes.GRAY2RGB ColorConversionCodes.GRAY2RGBA	그레이스케일 색상 공간을 BGR(A)나 RGB(A)로 변환
Py	cv2.COLOR_GRAY2BGR cv2.COLOR_GRAY2BGRA cv2.COLOR_GRAY2RGB cv2.COLOR_GRAY2RGBA	
C#	ColorConversionCodes.BGR2BGR565 ColorConversionCodes.BGRA2BGR565 ColorConversionCodes.RGB2BGR565 ColorConversionCodes.RGBA2BGR565	BGR(A)나 RGB(A) 색상 공간을 BGR565로 변환(BGR565 = 16비트)
Py	cv2.COLOR_BGR2BGR565 cv2.COLOR_BGRA2BGR565 cv2.COLOR_RGB2BGR565 cv2.COLOR_RGBA2BGR565	

C#	ColorConversionCodes.BGR5652BGR ColorConversionCodes.BGR5652BGRA ColorConversionCodes.BGR5652RGB ColorConversionCodes.BGR5652RGBA	BGR565 색상 공간을 BGR(A)나 RGB(A)로 변환(BGR565 = 16비트)
Py	cv2.COLOR_BGR5652BGR cv2.COLOR_BGR5652BGRA cv2.COLOR_BGR5652RGB cv2.COLOR_BGR5652RGBA	
C#	ColorConversionCodes.BGR5652GRAY ColorConversionCodes.GRAY2BGR565	BGR565와 그레이스케일 색상 공간 상호 변환(BGR565 = 16비트)
Py	cv2.COLOR_BGR5652GRAY cv2.COLOR_GRAY2BGR565	
C#	ColorConversionCodes.BGR2BGR555 ColorConversionCodes.BGRA2BGR555 ColorConversionCodes.RGB2BGR555 ColorConversionCodes.RGBA2BGR555	BGR(A)나 RGB(A) 색상 공간을 BGR555로 변환(BGR555 = 16비트)
Py	cv2.COLOR_BGR2BGR555 cv2.COLOR_BGRA2BGR555 cv2.COLOR_RGB2BGR555 cv2.COLOR_RGBA2BGR555	

C#

ColorConversionCodes.BGR5552BGR
ColorConversionCodes.BGR5552BGRA
ColorConversionCodes.BGR5552RGB
ColorConversionCodes.BGR5552RGBA

BGR555 색상 공간을 BGR(A)나 RGB(A)로 변환(BGR555 =
16비트)

Py

cv2.COLOR_BGR5552BGR
cv2.COLOR_BGR5552BGRA
cv2.COLOR_BGR5552RGB
cv2.COLOR_BGR5552RGBA

C#

ColorConversionCodes.BGR5552GRAY
ColorConversionCodes.GRAY2BGR555

BGR555와 그레이스케일 색상 공간 상호 변환(BGR555 =
16비트)

Py

cv2.COLOR_BGR5552GRAY
cv2.COLOR_GRAY2BGR555

C#

ColorConversionCodes.BGR2XYZ
ColorConversionCodes.RGB2XYZ
ColorConversionCodes.XYZ2BGR
ColorConversionCodes.XYZ2RGB

BGR이나 RGB 색상 공간을 CIE XYZ로 상호 변환(Rec. 709
색상 공간)

Py

cv2.COLOR_BGR2XYZ
cv2.COLOR_RGB2XYZ
cv2.COLOR_XYZ2BGR
cv2.COLOR_XYZ2RGB

C#	ColorConversionCodes.BGR2YCrCb	BGR0 나 RGB 색상 공간을 YCC(크로마)로 상호 변환(YCC = Y, Cr, Cb)
	ColorConversionCodes.RGB2YCrCb	
	ColorConversionCodes.YCrCb2BGR	
	ColorConversionCodes.YCrCb2RGB	
Py	cv2.COLOR_BGR2YCrCb	BGR0 나 RGB 색상 공간을 YCC(크로마)로 상호 변환(YCC = Y, Cr, Cb)
	cv2.COLOR_RGB2YCrCb	
	cv2.COLOR_YCrCb2BGR	
	cv2.COLOR_YCrCb2RGB	
C#	ColorConversionCodes.BGR2HSV	BGR0 나 RGB 색상 공간을 HSV로 상호 변환(HSV = Hue, Saturation, Value)
	ColorConversionCodes.RGB2HSV	
	ColorConversionCodes.HSV2BGR	
	ColorConversionCodes.HSV2RGB	
Py	cv2.COLOR_BGR2HSV	BGR0 나 RGB 색상 공간을 HSV로 상호 변환(HSV = Hue, Saturation, Value)
	cv2.COLOR_RGB2HSV	
	cv2.COLOR_HSV2BGR	
	cv2.COLOR_HSV2RGB	
C#	ColorConversionCodes.BGR2HLS	BGR0 나 RGB 색상 공간을 HLS로 상호 변환(HLS = Hue, Lightness, Saturation)
	ColorConversionCodes.RGB2HLS	
	ColorConversionCodes.HLS2BGR	
	ColorConversionCodes.HLS2RGB	
Py	cv2.COLOR_BGR2HLS	BGR0 나 RGB 색상 공간을 HLS로 상호 변환(HLS = Hue, Lightness, Saturation)
	cv2.COLOR_RGB2HLS	
	cv2.COLOR_HLS2BGR	
	cv2.COLOR_HLS2RGB	

C#	ColorConversionCodes.BGR2Lab ColorConversionCodes.RGB2Lab ColorConversionCodes.Lab2BGR ColorConversionCodes.Lab2RGB	BGR0이나 RGB 색상 공간을 CIE Lab으로 상호 변환(Lab = 반 사율, 색도1, 색도2)
Py	cv2.COLOR_BGR2Lab cv2.COLOR_RGB2Lab cv2.COLOR_Lab2BGR cv2.COLOR_Lab2RGB	
C#	ColorConversionCodes.BGR2Luv ColorConversionCodes.RGB2Luv ColorConversionCodes.Luv2BGR ColorConversionCodes.Luv2RGB	BGR0이나 RGB 색상 공간을 CIE Luv로 상호 변환(CIE UVW 기반)
Py	cv2.COLOR_BGR2Luv cv2.COLOR_RGB2Luv cv2.COLOR_Luv2BGR cv2.COLOR_Luv2RGB	
C#	ColorConversionCodes.BGR2YUV ColorConversionCodes.RGB2YUV ColorConversionCodes.YUV2BGR ColorConversionCodes.YUV2RGB	BGR0이나 RGB 색상 공간을 YUV로 상호 변환(YUV = 밝기, 색 상1, 색상2)
Py	cv2.COLOR_BGR2YUV cv2.COLOR_RGB2YUV cv2.COLOR_YUV2BGR cv2.COLOR_YUV2RGB	

$$Y = 0.299 \times R + 0.587 \times G + 0.114 \times B$$

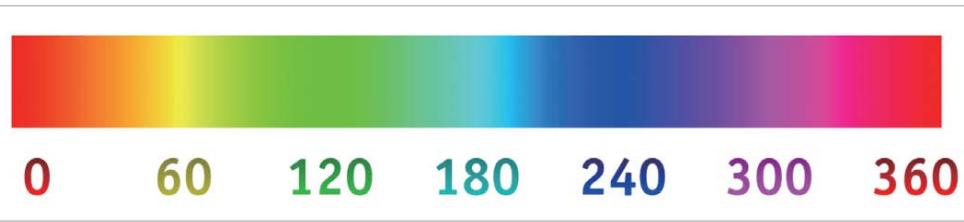


그림 5.1 일반적인 Hue의 범위

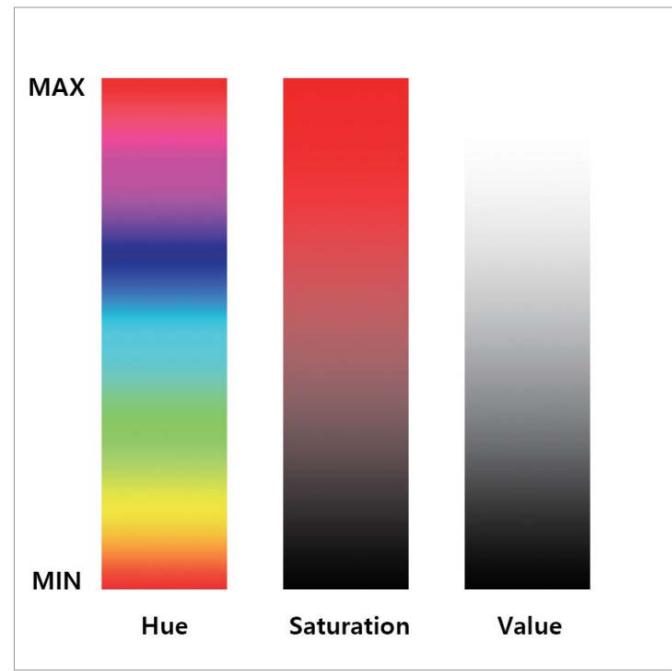


그림 5.2 HSV 색상 모델

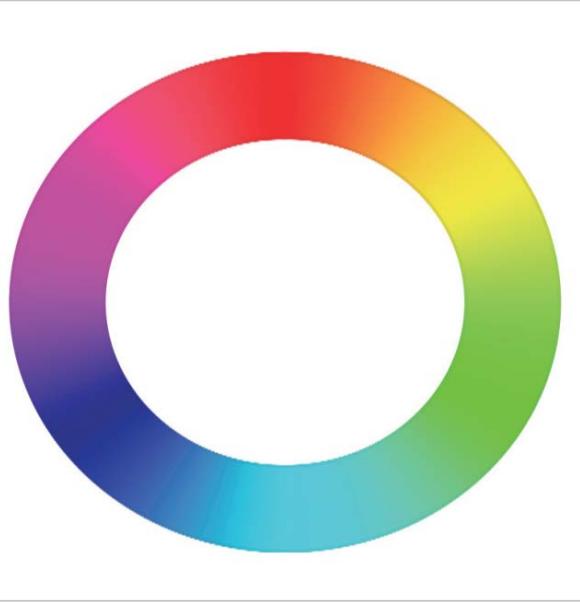


그림 5.3 Hue 원형 모델

예제 5.3 C# OpenCvSharp에서의 Hue 공간 색상 검출

```
using System;
using OpenCvSharp;

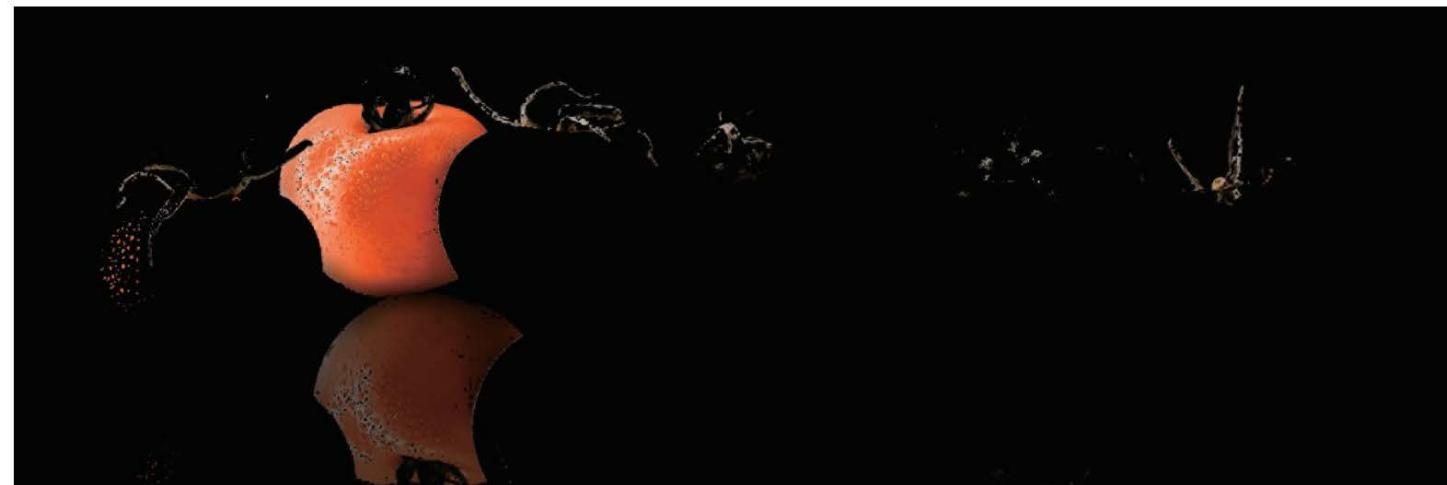
namespace Project
{
    class Program
    {
        static void Main(string[] args)
        {
            Mat src = Cv2.ImRead("tomato.jpg");
            Mat hsv = new Mat(src.Size(), MatType.CV_8UC3);
            Mat dst = new Mat(src.Size(), MatType.CV_8UC3);

            Cv2.CvtColor(src, hsv, ColorConversionCodes.BGR2HSV);
            Mat[] HSV = Cv2.Split(hsv);
            Mat H_orange = new Mat(src.Size(), MatType.CV_8UC1);
            Cv2.InRange(HSV[0], new Scalar(8), new Scalar(20), H_orange);

            Cv2.BitwiseAnd(hsv, hsv, dst, H_orange);
            Cv2.CvtColor(dst, dst, ColorConversionCodes.HSV2BGR);

            Cv2.ImShow("Orange", dst);
            Cv2.WaitKey(0);
            Cv2.DestroyAllWindows();
        }
    }
}
```

[출력 결과]



예제 5.4 Python OpenCV에서의 Hue 공간 색상 검출

```
import cv2

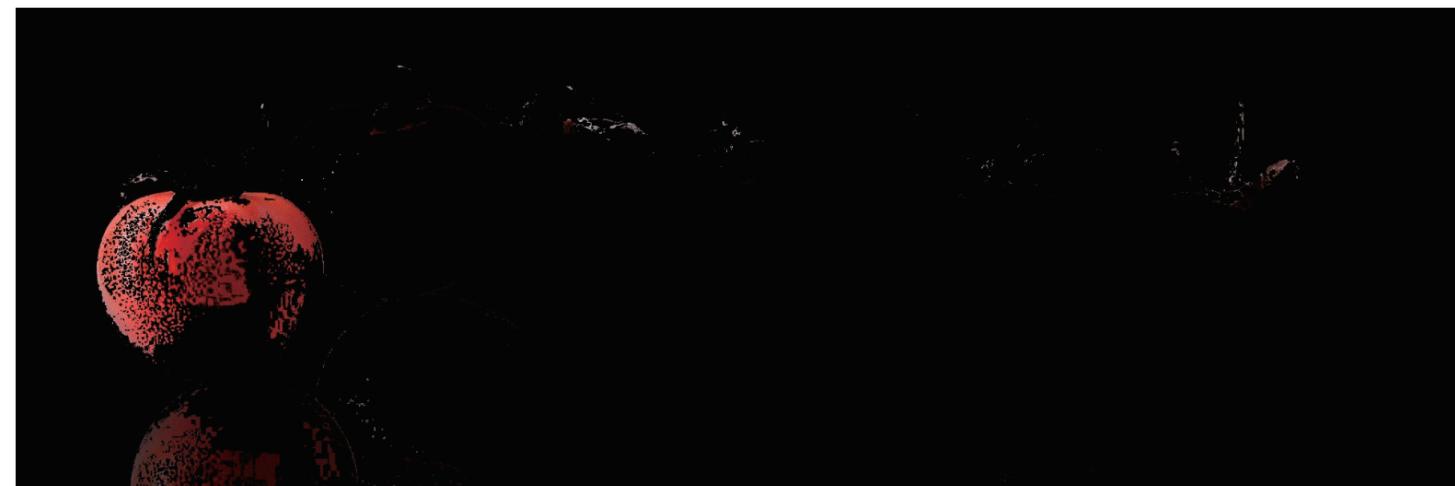
src = cv2.imread("tomato.jpg")
hsv = cv2.cvtColor(src, cv2.COLOR_BGR2HSV)

h, s, v = cv2.split(hsv)
h_red = cv2.inRange(h, 0, 5)

dst = cv2.bitwise_and(hsv, hsv, mask = h_red)
dst = cv2.cvtColor(dst, cv2.COLOR_HSV2BGR)

cv2.imshow("dst", dst)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

[출력 결과]



배열 병합 함수 수식:

$$\text{dst} = \text{src1} \times \text{alpha} + \text{src2} \times \text{beta} + \text{gamma}$$

예제 5.5 C# OpenCvSharp에서의 색상 검출

```
using System;
using OpenCvSharp;

namespace Project
{
    class Program
    {
        static void Main(string[] args)
        {
            Mat src = Cv2.ImRead("tomato.jpg");
            Mat hsv = new Mat(src.Size(), MatType.CV_8UC3);
            Mat lower_red = new Mat(src.Size(), MatType.CV_8UC3);
            Mat upper_red = new Mat(src.Size(), MatType.CV_8UC3);
            Mat added_red = new Mat(src.Size(), MatType.CV_8UC3);
            Mat dst = new Mat(src.Size(), MatType.CV_8UC3);

            Cv2.CvtColor(src, hsv, ColorConversionCodes.BGR2HSV);

            Cv2.InRange(hsv, new Scalar(0, 100, 100), new Scalar(5, 255, 255), lower_red);
            Cv2.InRange(hsv, new Scalar(170, 100, 100), new Scalar(179, 255, 255), upper_red);
            Cv2.AddWeighted(lower_red, 1.0, upper_red, 1.0, 0.0, added_red);

            Cv2.BitwiseAnd(hsv, hsv, dst, added_red);
            Cv2.CvtColor(dst, dst, ColorConversionCodes.HSV2BGR);
```

```
Cv2.ImShow("dst", dst);
Cv2.WaitKey(0);
Cv2.DestroyAllWindows();
}
}
}
```

[출력 결과]



예제 5.6 Python OpenCV에서의 색상 검출

```
import cv2

src = cv2.imread("tomato.jpg")
hsv = cv2.cvtColor(src, cv2.COLOR_BGR2HSV)

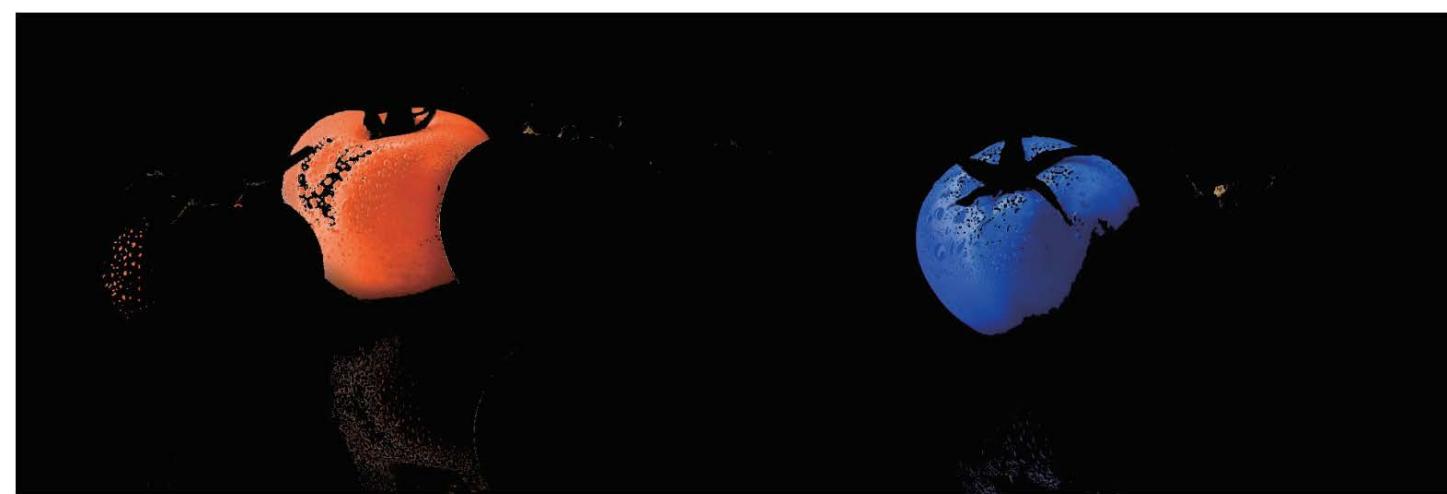
h, s, v = cv2.split(hsv)

orange = cv2.inRange(hsv, (8, 100, 100), (20, 255, 255))
blue = cv2.inRange(hsv, (110, 100, 100), (130, 255, 255))
mix_color = cv2.addWeighted(orange, 1.0, blue, 1.0, 0.0)

dst = cv2.bitwise_and(hsv, hsv, mask = mix_color)
dst = cv2.cvtColor(dst, cv2.COLOR_HSV2BGR)

cv2.imshow("dst", dst)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

【출력 결과】



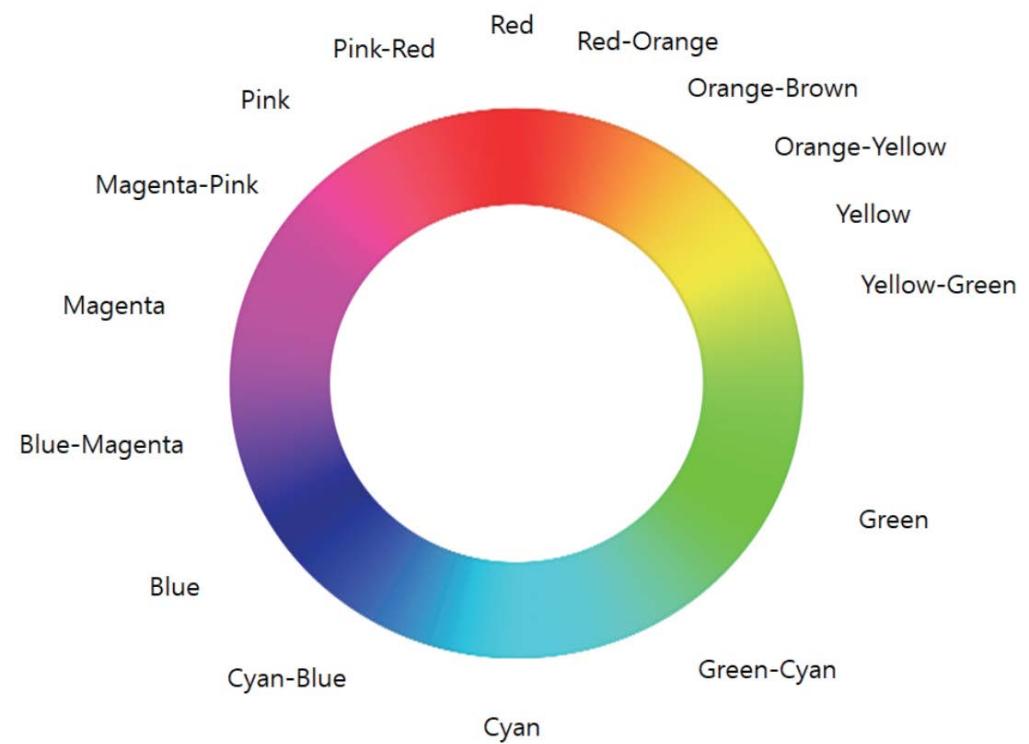


그림 5.4 Hue 원형 색상 표기 모델

표 5.2 Hue 색상 파라미터

색상	값	색상	값
Red	178~5	Cyan	85~100
Red–Orange	6~10	Cyan–Blue	101~110
Orange–Brown	11~20	Blue	110~120
Orange–Yellow	21~25	Blue–Magenta	121~140
Yellow	26~30	Magenta	141~160
yellow–Green	31~40	Magenta–Pink	161~165
Green	41~70	Pink	166~173
Green–Cyan	71~84	Pink–Red	174~177

표 5.3 HSV 색상 파라미터

색상	Hue	Saturation	Value
White	0	0	255
Sliver	0	0	192
Gray	0	0	128
Black	0	0	0
Red	0	255	255
Maroon	0	255	128
Orange	15	255	255
Brown	15	255	128
Yellow	30	255	255
Olive	30	255	128
Lime	60	255	255
Green	60	255	128
Aqua	90	255	255
Teal	90	255	128
Blue	120	255	255
Navy	120	255	128
Magenta	150	255	255
Purple	150	255	128
Pink	150	128	255

예제 5.7 C# OpenCvSharp에서의 이진화

```
using System;
using OpenCvSharp;

namespace Project
{
    class Program
    {
        static void Main(string[] args)
        {
            Mat src = Cv2.ImRead("swan.jpg");
            Mat gray = new Mat(src.Size(), MatType.CV_8UC1);
            Mat binary = new Mat(src.Size(), MatType.CV_8UC1);

            Cv2.CvtColor(src, gray, ColorConversionCodes.BGR2GRAY);
            Cv2.Threshold(gray, binary, 127, 255, ThresholdTypes.Otsu);

            Cv2.ImShow("binary", binary);
            Cv2.WaitKey(0);
            Cv2.DestroyAllWindows();
        }
    }
}
```

[출력 결과]



예제 5.8 Python OpenCV에서의 이진화

```
import cv2

src = cv2.imread("swan.jpg")
_, binary = cv2.threshold(src, 127, 255, cv2.THRESH_BINARY)

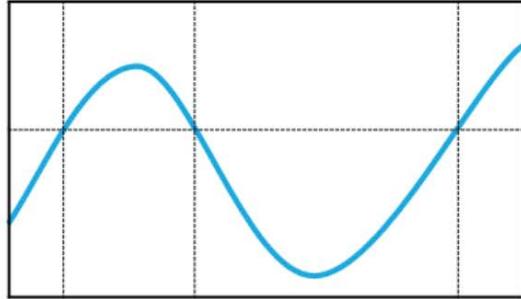
cv2.imshow("binary", binary)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

[출력 결과]

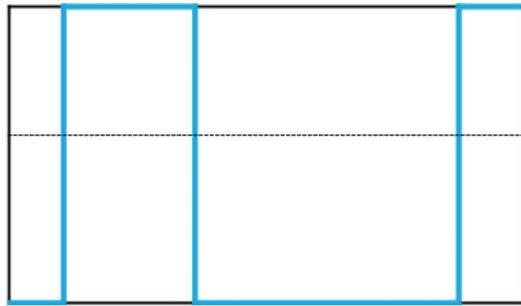


표 5.4 임곗값 형식

언어	속성	설명
C#	ThresholdTypes.Binary	$dst = (src > thresh) ? maxval : 0$
Py	cv2.THRESH_BINARY	(임곗값을 초과할 경우 maxval, 아닐 경우 0)
C#	ThresholdTypes.BinaryInv	$dst = (src > thresh) ? 0 : maxval$
Py	cv2.THRESH_BINARY_INV	(임곗값을 초과할 경우 0, 아닐 경우 maxval)
C#	ThresholdTypes.Trunc	$dst = (src > thresh) ? thresh : src$
Py	cv2.THRESH_TRUNC	(임곗값을 초과할 경우 thresh, 아닐 경우 변형 없음)
C#	ThresholdTypes.Tozero	$dst = (src > thresh) ? src : 0$
Py	cv2.THRESH_TOZERO	(임곗값을 초과할 경우 변형 없음, 아닐 경우 0)
C#	ThresholdTypes.TozeroInv	$dst = (src > thresh) ? 0 : src$
Py	cv2.THRESH_TOZERO_INV	(임곗값을 초과할 경우 0, 아닐 경우 변형 없음)
C#	ThresholdTypes.Mask	검은색 이미지로 변경(마스크용)
Py	cv2.THRESH_MASK	
C#	ThresholdTypes.Otsu	오츠 알고리즘 적용(단일 채널 이미지에만 적용 가능)
Py	cv2.THRESH_OTSU	
C#	ThresholdTypes.Triangle	삼각형(Triangle) 알고리즘 적용(단일 채널 이미지에만 적용 가능)
Py	cv2.THRESH_TRIANGLE	

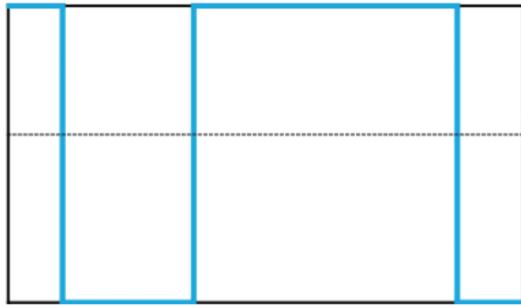


원본 이미지(src)



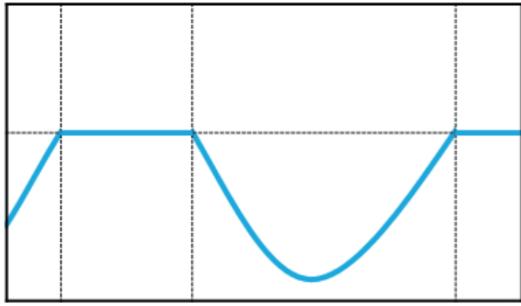
C# ThresholdTypes.Binary

Py cv2.THRESH_BINARY



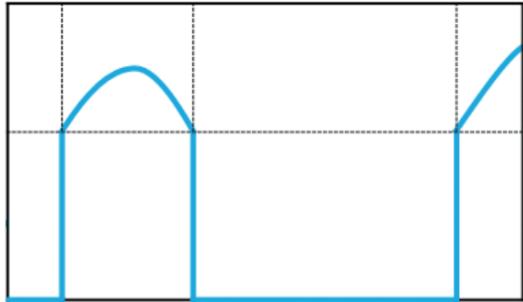
C# ThresholdTypes.BinaryInv

Py cv2.THRESH_BINARY_INV



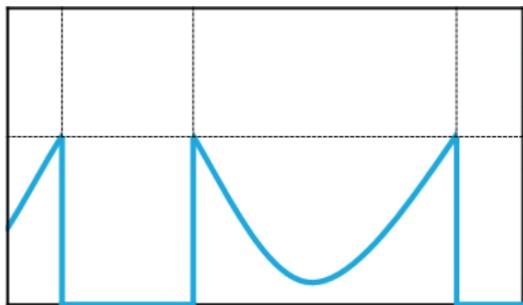
C# ThresholdTypes.Trunc

Py cv2.THRESH_TRUNC



C# ThresholdTypes.Tozero

Py cv2.THRESH_TOZERO



C# ThresholdTypes.TozeroInv

Py cv2.THRESH_TOZERO_INV

그림 5.5 이진화 함수의 임곗값 형식에 따른 결과 유형

$$\sigma^2 \equiv \alpha \times \sigma_1^2 + \beta \times \sigma_2^2$$

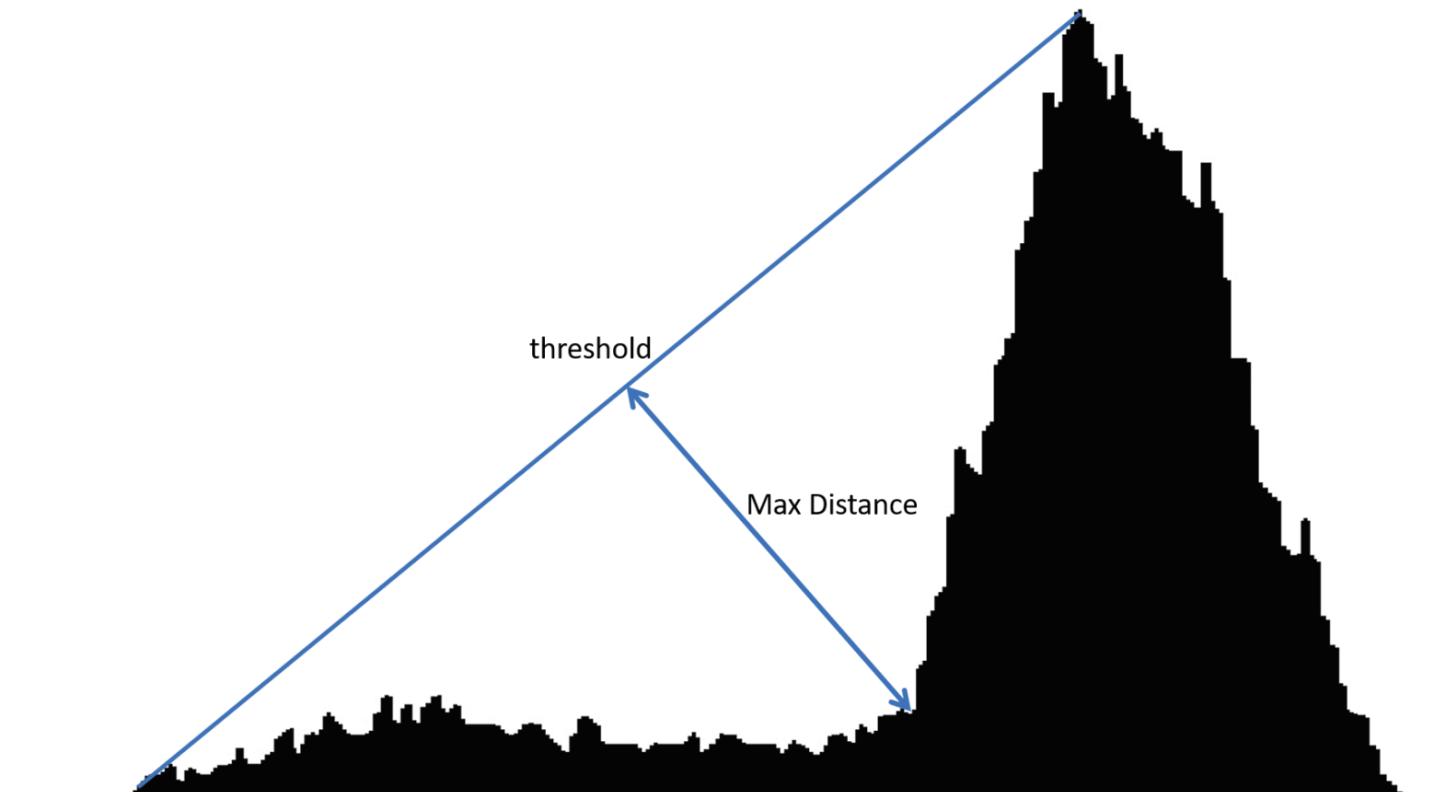


그림 5.6 삼각형 알고리즘의 연산 방법

$$T(x,y) = \frac{1}{blocksize^2} \sum_{x_i}\sum_{y_i} I(x+x_i,y+y_i) - c$$

표 5.5 적응형 이진화 플래그

언어	속성	설명
C#	AdaptiveThresholdTypes.MeanC	
Py	cv2.ADAPTIVE_THRESH_MEAN_C	blockSize 영역의 모든 픽셀에 평균 가중치를 적용
C#	AdaptiveThresholdTypes.GaussianC	blockSize 영역의 모든 픽셀에 중심점으로부터의 거리에 대한 가우시안 가중치를 적용
Py	cv2.ADAPTIVE_THRESH_GAUSSIAN_C	우시안 가중치를 적용

예제 5.8 C# OpenCvSharp에서의 적응형 이진화

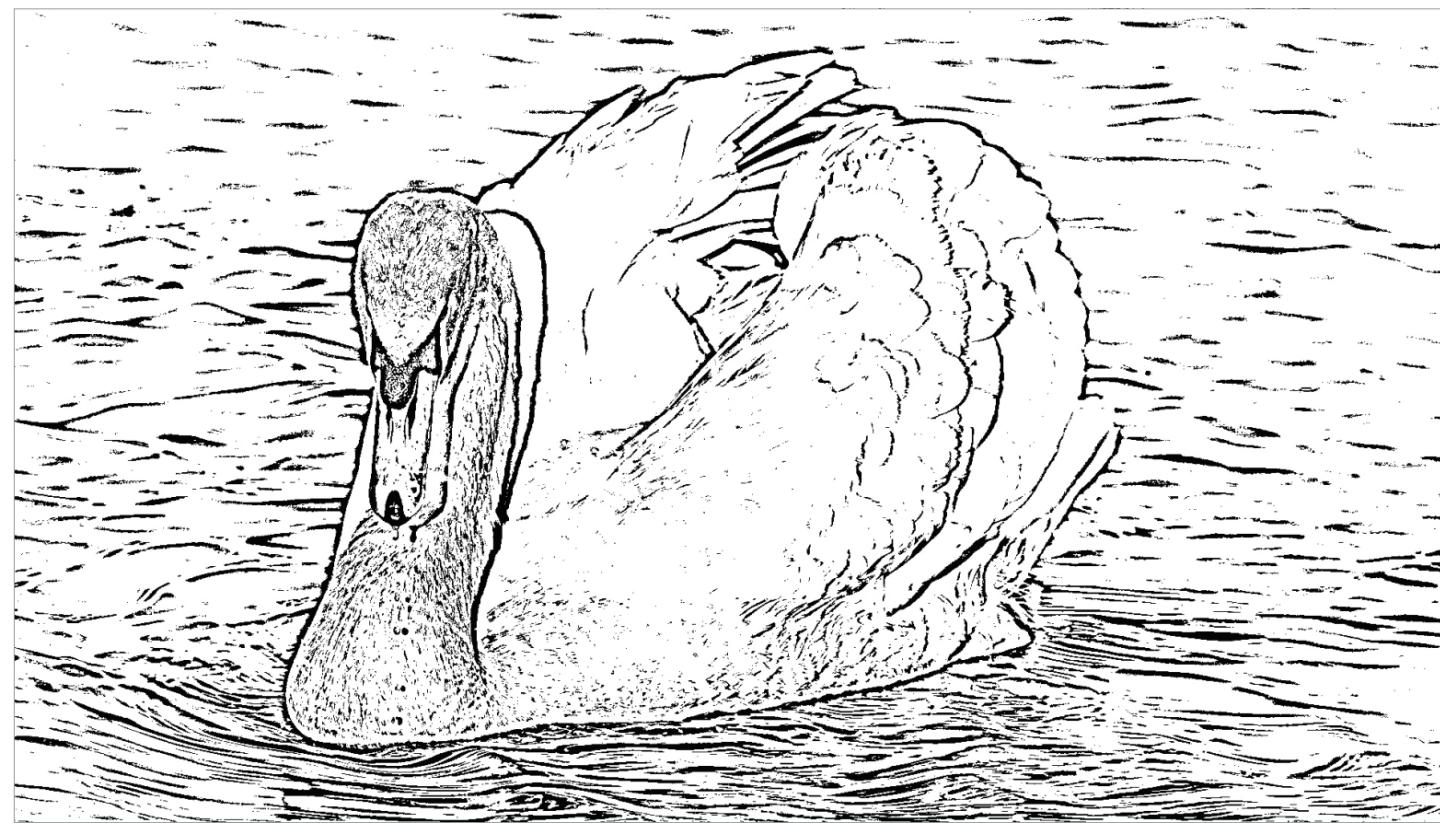
```
using System;
using OpenCvSharp;

namespace Project
{
    class Program
    {
        static void Main(string[] args)
        {
            Mat src = Cv2.ImRead("swan.jpg");
            Mat gray = new Mat(src.Size(), MatType.CV_8UC1);
            Mat binary = new Mat(src.Size(), MatType.CV_8UC1);

            Cv2.CvtColor(src, gray, ColorConversionCodes.BGR2GRAY);
            Cv2.AdaptiveThreshold(gray, binary, 255, AdaptiveThresholdTypes.GaussianC,
            ThresholdTypes.Binary, 25, 5);

            Cv2.ImShow("binary", binary);
            Cv2.WaitKey(0);
            Cv2.DestroyAllWindows();
        }
    }
}
```

[출력 결과]



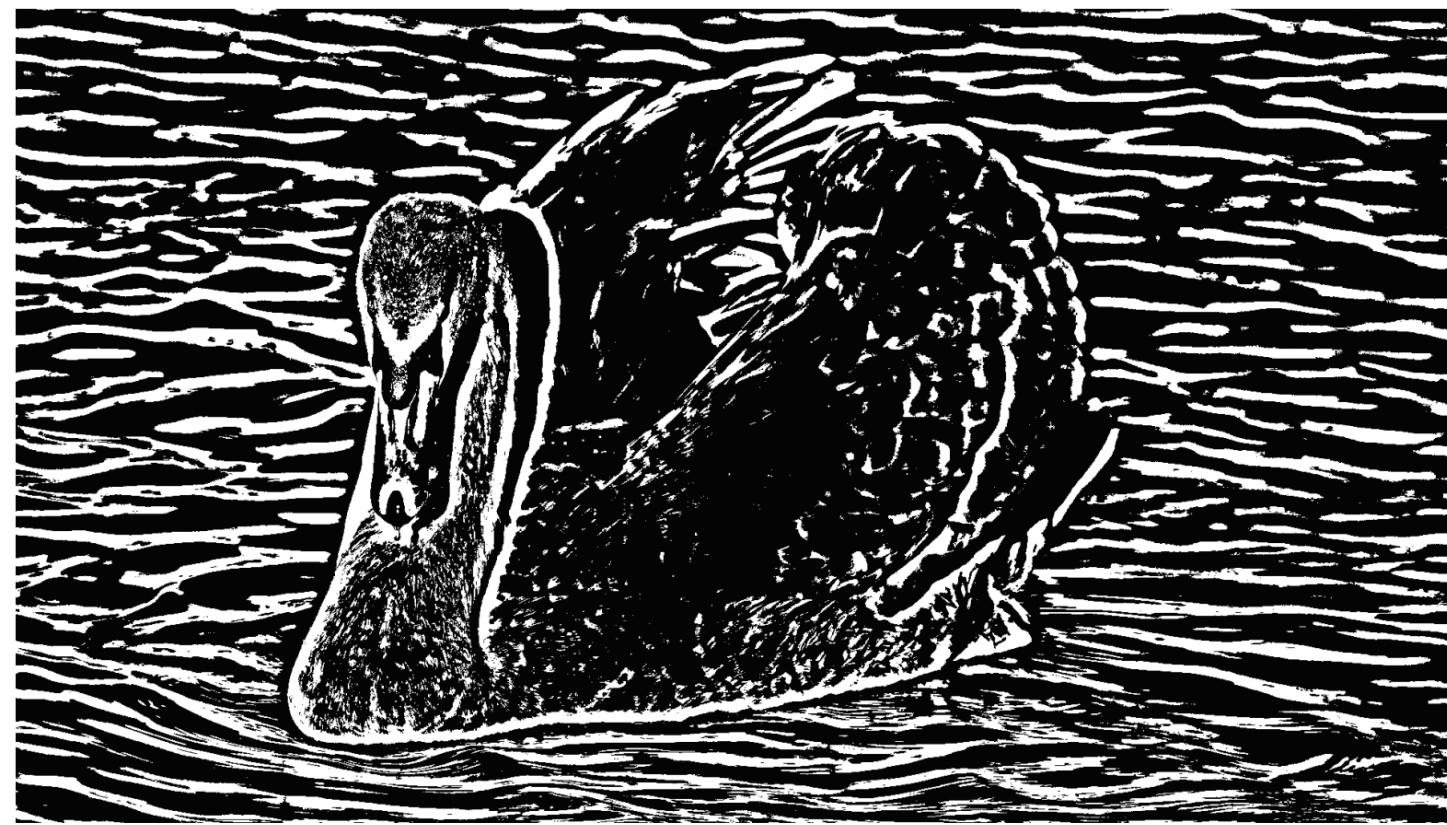
예제 5.9 Python OpenCV에서의 적응형 이진화

```
import cv2

src = cv2.imread("swan.jpg")
gray = cv2.cvtColor(src, cv2.COLOR_BGR2GRAY)
binary = cv2.adaptiveThreshold(gray, 255, cv2.ADAPTIVE_THRESH_MEAN_C, cv2.THRESH_BINARY, 33, -5)

cv2.imshow("binary", binary)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

[출력 결과]



두 배열의 크기와 채널 수가 같은 경우 두 배열의 요소별 합:

$$\text{dst}(\mathbf{I}) = \text{src1}(\mathbf{I}) + \text{src2}(\mathbf{I}) \quad \text{if } \text{mask}(\mathbf{I}) \neq 0$$

배열과 스칼라의 요소별 합:

$$\text{dst}(\mathbf{I}) = \text{src1}(\mathbf{I}) + \text{src2} \quad \text{if } \text{mask}(\mathbf{I}) \neq 0$$

스칼라와 배열의 요소별 합:

$$\text{dst}(\mathbf{I}) = \text{src1} + \text{src2}(\mathbf{I}) \quad \text{if } \text{mask}(\mathbf{I}) \neq 0$$

두 배열의 크기와 채널 수가 같은 경우 두 배열의 요소별 차:

$$dst(I) = src1(I) - src2(I) \quad if \ mask(I) \neq 0$$

$$dst(I) = src2(I) - src1(I) \quad if \ mask(I) \neq 0$$

배열과 스칼라의 요소별 차:

$$\text{dst}(\mathbf{I}) = \text{src1} - \text{src2}(\mathbf{I}) \quad \text{if } \text{mask}(\mathbf{I}) \neq 0$$

스칼라와 배열의 요소별 차:

$$\text{dst}(\mathbf{I}) = \text{src1}(\mathbf{I}) - \text{src2} \quad \text{if } \text{mask}(\mathbf{I}) \neq 0$$

두 배열의 크기와 채널 수가 같은 경우 두 배열의 요소별 곱:

$$dst(I) = src1(I) \times src2(I) \times scale$$

배열과 스칼라의 요소별 곱:

$$dst(I) = src1 \times src2(I) \times scale$$

스칼라와 배열의 요소별 곱:

$$dst(I) = src1(I) \times src2 \times scale$$

두 배열의 크기와 채널 수가 같은 경우 두 배열의 요소별 나눔:

$$\text{dst}(\mathbf{I}) = \text{src1}(\mathbf{I}) \div \text{src2}(\mathbf{I}) \times \text{scale}$$

배열과 스칼라의 요소별 나눔:

$$\text{dst}(\mathbf{I}) = \text{src1} \div \text{src2}(\mathbf{I}) \times \text{scale}$$

스칼라와 배열의 요소별 나눔:

$$\text{dst}(\mathbf{I}) = \text{src1}(\mathbf{I}) \div \text{src2} \times \text{scale}$$

비율과 배열의 요소별 나눔:

$$\text{dst}(\text{I}) = \text{scale} \div \text{src2}(\text{I})$$

두 배열의 크기와 채널 수가 같은 경우 두 배열의 요소별 최댓값:

$$\text{dst}(\mathbf{I}) = \max(\text{src1}(\mathbf{I}), \text{src2}(\mathbf{I}))$$

배열과 스칼라의 요소별 최댓값:

$$\text{dst}(\mathbf{I}) = \max(\text{src1}(\mathbf{I}), \text{src2})$$

스칼라와 배열의 요소별 최댓값:

$$dst(I) = \max(src1, src2(I))$$

배열과 부동 소수점의 요소별 최댓값

$$dst(I) = \max(src1, value)$$

두 배열의 크기와 채널 수가 같은 경우 두 배열의 요소별 최솟값:

$$\text{dst}(\mathbf{I}) = \min(\text{src1}(\mathbf{I}), \text{src2}(\mathbf{I}))$$

배열과 스칼라의 요소별 최솟값:

$$dst(I) = \min(src1(I), src2)$$

스칼라와 배열의 요소별 최솟값:

$$\text{dst}(\text{I}) = \min(\text{src1}, \text{src2}(\text{I}))$$

배열과 부동 소수점의 요소별 최솟값:

$$\text{dst}(\text{I}) = \min(\text{src1}, \text{value})$$

배열의 요소별 절댓값:

$$\text{dst}(\mathbf{I}) = \text{abs}(|\text{src}|)$$

두 배열의 크기와 채널 수가 같은 경우 두 배열의 요소별 절댓값 차이:

$$\text{dst}(\text{I}) = \text{abs}(|\text{src1}(\text{I}) - \text{src2}(\text{I})|)$$

배열과 스칼라의 요소별 절댓값 차이:

$$dst(I) = \text{abs}(|\text{src1}(I) - \text{src2}|)$$

스칼라와 배열의 요소별 절댓값 차이:

$$dst(I) = \text{abs}(|\text{src1} - \text{src2}(I)|)$$

표 5.6 비교 함수 플래그

언어	속성	설명
C#	CmpTypes.EQ	src1과 src2의 요소가 같음
Py	cv2.CMP_EQ	
C#	CmpTypes.GT	src10 src2보다 요소가 큼
Py	cv2.CMP_GT	
C#	CmpTypes.GE	src10 src2보다 요소가 크거나 같음
Py	cv2.CMP_GE	
C#	CmpTypes.LT	src10 src2보다 요소가 작음
Py	cv2.CMP_LT	
C#	CmpTypes.LE	src10 src2보다 요소가 작거나 같음
Py	cv2.CMP_LE	
C#	CmpTypes.NE	src1과 src2의 요소가 같지 않음
Py	cv2.CMP_NE	

두 배열의 크기와 채널 수가 같은 경우 두 배열의 요소별 비교:

$\text{dst}(\text{I}) = \text{src1}(\text{I}) \text{ cmpop } \text{src2}(\text{I})$

배열과 스칼라의 요소별 비교:

$\text{dst}(\text{I}) = \text{src1}(\text{I}) \text{ cmpop } \text{src2}$

스칼라와 배열의 요소별 비교:

$\text{dst}(\text{I}) = \text{src1 cmpop src2}(\text{I})$

선형 방정식 시스템의 해 찾기 수식:

$$\text{dst} = \operatorname{argmin}_x \|A \cdot X - B\|$$

표 5.7 선형 방정식 해 찾기 함수 플래그

언어	속성	설명
C#	DecompTypes.LU	LU 분해(가우스 소거)
Py	cv2.DECOMP_LU	
C#	DecompTypes.SVD	특잇값 분해
Py	DECOMP_SVD	
C#	DecompTypes.Eig	고윳값 분해(src1 행렬은 대칭이어야 함)
Py	DECOMP_EIG	
C#	DecompTypes.Cholesky	솔레스키 분해(src1 행렬은 대칭이어야 함)
Py	DECOMP_CHOLESKY	
C#	DecompTypes.QR	QR 인수 분해
Py	DECOMP_QR	
C#	DecompTypes.Normal	노멀 분해(src1 · dst = src2 방정식 사용)
Py	DECOMP_NORMAL	

두 배열의 크기와 채널 수가 같은 경우 두 배열의 요소별 논리곱:

$$\text{dst}(\mathbf{I}) = \text{src1}(\mathbf{I}) \wedge \text{src2}(\mathbf{I}) \quad \text{if } \text{mask}(\mathbf{I}) \neq 0$$

배열과 스칼라의 요소별 논리곱:

$$\text{dst}(\mathbf{I}) = \text{src1}(\mathbf{I}) \wedge \text{src2} \quad \text{if } \text{mask}(\mathbf{I}) \neq 0$$

스칼라와 배열의 요소별 논리곱:

$$\text{dst}(\mathbf{I}) = \text{src1} \wedge \text{src2}(\mathbf{I}) \quad \text{if } \text{mask}(\mathbf{I}) \neq 0$$

두 배열의 크기와 채널 수가 같은 경우 두 배열의 요소별 논리합:

$$\text{dst}(\mathbf{I}) = \text{src1}(\mathbf{I}) \vee \text{src2}(\mathbf{I}) \quad \text{if } \text{mask}(\mathbf{I}) \neq 0$$

배열과 스칼라의 요소별 논리합:

$$\text{dst}(\mathbf{I}) = \text{src1}(\mathbf{I}) \vee \text{src2} \quad \text{if } \text{mask}(\mathbf{I}) \neq 0$$

스칼라와 배열의 요소별 논리합:

$$\text{dst}(\mathbf{I}) = \text{src1} \vee \text{src2}(\mathbf{I}) \quad \text{if } \text{mask}(\mathbf{I}) \neq 0$$

두 배열의 크기와 채널 수가 같은 경우 두 배열의 요소별 논리합:

$$\text{dst}(\mathbf{I}) = \text{src1}(\mathbf{I}) \oplus \text{src2}(\mathbf{I}) \quad \text{if } \text{mask}(\mathbf{I}) \neq 0$$

배열과 스칼라의 요소별 논리합:

$$\text{dst}(\mathbf{I}) = \text{src1}(\mathbf{I}) \oplus \text{src2} \quad \text{if } \text{mask}(\mathbf{I}) \neq 0$$

스칼라와 배열의 요소별 논리합:

$$\text{dst}(I) = \text{src1} \oplus \text{src2}(I) \quad \text{if } \text{mask}(I) \neq 0$$

배열의 요소별 반전:

$\text{dst}(I) = \sim \text{src} \quad \text{if } \text{mask}(I) \neq 0$

스칼라의 요소별 반전:

$$\text{dst}(\mathbf{I}) = \sim \text{src}(\mathbf{I}) \quad \text{if } \text{mask}(\mathbf{I}) \neq 0$$

예제 5.11 C# OpenCvSharp 비교 함수

```
using System;
using OpenCvSharp;

namespace Project
{
    class Program
    {
        static void Main(string[] args)
        {
            Mat src1 = Cv2.ImRead("gerbera.jpg");
            Mat dst = new Mat(src1.Size(), MatType.CV_8UC3);

            Cv2.Compare(src1, new Scalar(200, 127, 100), dst, CmpTypes.GT);

            Cv2.ImShow("dst", dst);
            Cv2.WaitKey(0);
            Cv2.DestroyAllWindows();
        }
    }
}
```

[출력 결과]



예제 5.12 Python OpenCV의 선형 방정식 시스템의 해 찾기 함수

```
import cv2
import numpy as np

src1 = np.array([[9, 2], [1, 1]], dtype=np.double)
src2 = np.array([38, 5], dtype=np.double)

dst = cv2.solve(src1, src2, flags=cv2.DECOMP_LU)
print(dst)
```

[출력 결과]

```
(True, array([[4.],  
             [1.]]))
```

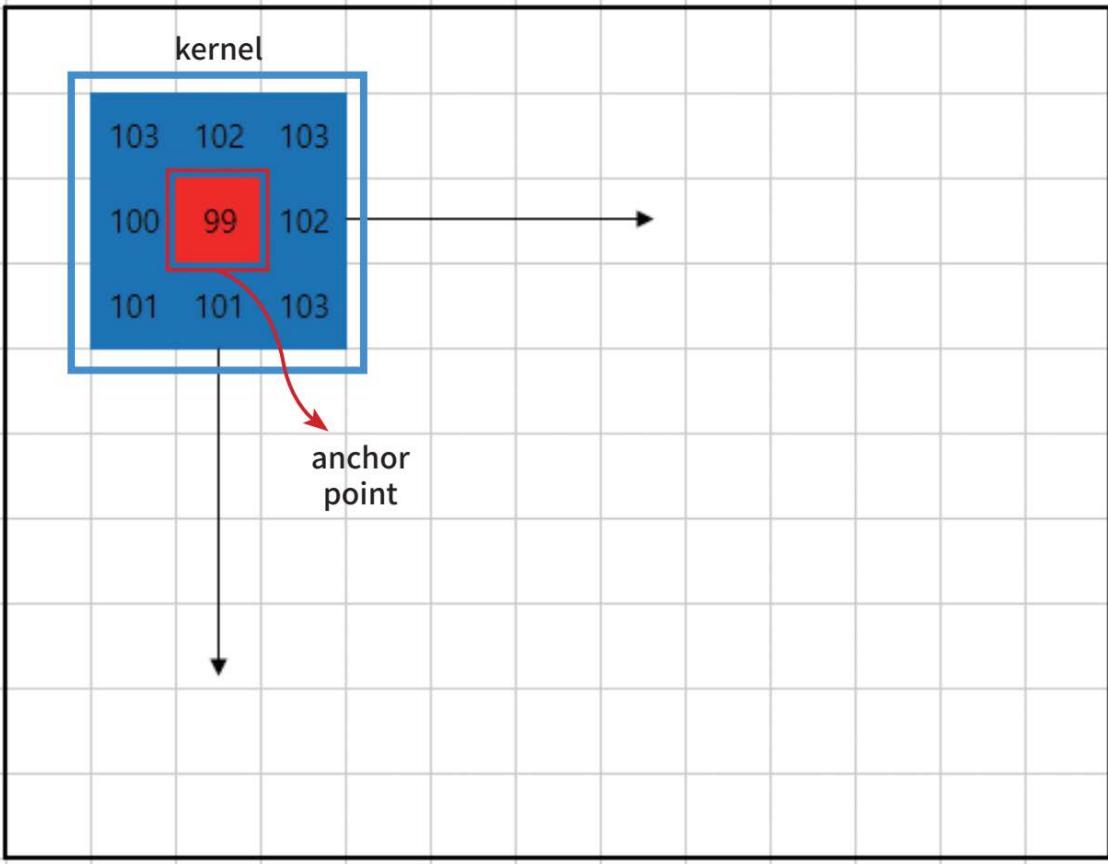


그림 5.7 커널과 고정점

표 5.8 테두리 유형

언어	속성	설명
C#	BorderTypes.Constant	고정 값으로 픽셀을 확장 iiiiii abcdefgh iiuiii
Py	cv2.BORDER_CONSTANT	
C#	BorderTypes.Replicate	테두리 픽셀을 복사해서 확장 aaaaaaaa abcdefgh hhhhhh
Py	cv2.BORDER_REPLICATE	
C#	BorderTypes.Reflect	픽셀을 반사해서 확장 fedcba abcdefgh hgfedcb
Py	cv2.BORDER_REFLECT	
C#	BorderTypes.Wrap	반대쪽 픽셀을 복사해서 확장 cdefgh abcdefgh abcdefg
Py	cv2.BORDER_WRAP	
C#	BorderTypesReflect101	이중 픽셀을 만들지 않고 반사해서 확장 gfedcb abcdefgh gfedcba
Py	cv2.BORDER_REFLECT_101	
C#	BorderTypes.Default	Reflect101 방식을 사용
Py	cv2.BORDER_DEFAULT	
C#	BorderTypes.Transparent	픽셀을 투명하게 해서 확장 uvwxyz abcdefgh ijklmno
Py	cv2.BORDER_TRANSPARENT	
C#	BorderTypes.Isolated	관심 영역(ROI) 밖은 고려하지 않음
Py	cv2.BORDER_ISOLATED	

$\frac{1}{25}$

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

그림 5.8 정규화된 박스 필터

$$\sigma_x = 0.3 \times \left(\left(\frac{ksize.width - 1}{2} \right) - 1 \right) + 0.8$$

$$\sigma_y = 0.3 \times \left(\left(\frac{ksize.height - 1}{2} \right) - 1 \right) + 0.8$$

예제 5.13 C# OpenCvSharp에서의 가우시안 흐림 효과 함수

```
using System;
using OpenCvSharp;

namespace Project
{
    class Program
    {
        static void Main(string[] args)
        {
            Mat src = Cv2.ImRead("crescent.jpg");
            Mat dst = new Mat(src.Size(), MatType.CV_8UC3);

            Cv2.GaussianBlur(src, dst, new Size(9, 9), 3, 3, BorderTypes.Isolated);

            Cv2.ImShow("dst", dst);
            Cv2.WaitKey(0);
            Cv2.DestroyAllWindows();
        }
    }
}
```

[출력 결과]



예제 5.14 Python OpenCV에서의 양방향 필터 흐림 효과 함수

```
import cv2

src = cv2.imread("crescent.jpg")

dst = cv2.bilateralFilter(src, 100, 33, 11, borderType=cv2.BORDER_ISOLATED)

cv2.imshow("dst", dst)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

【출력 결과】



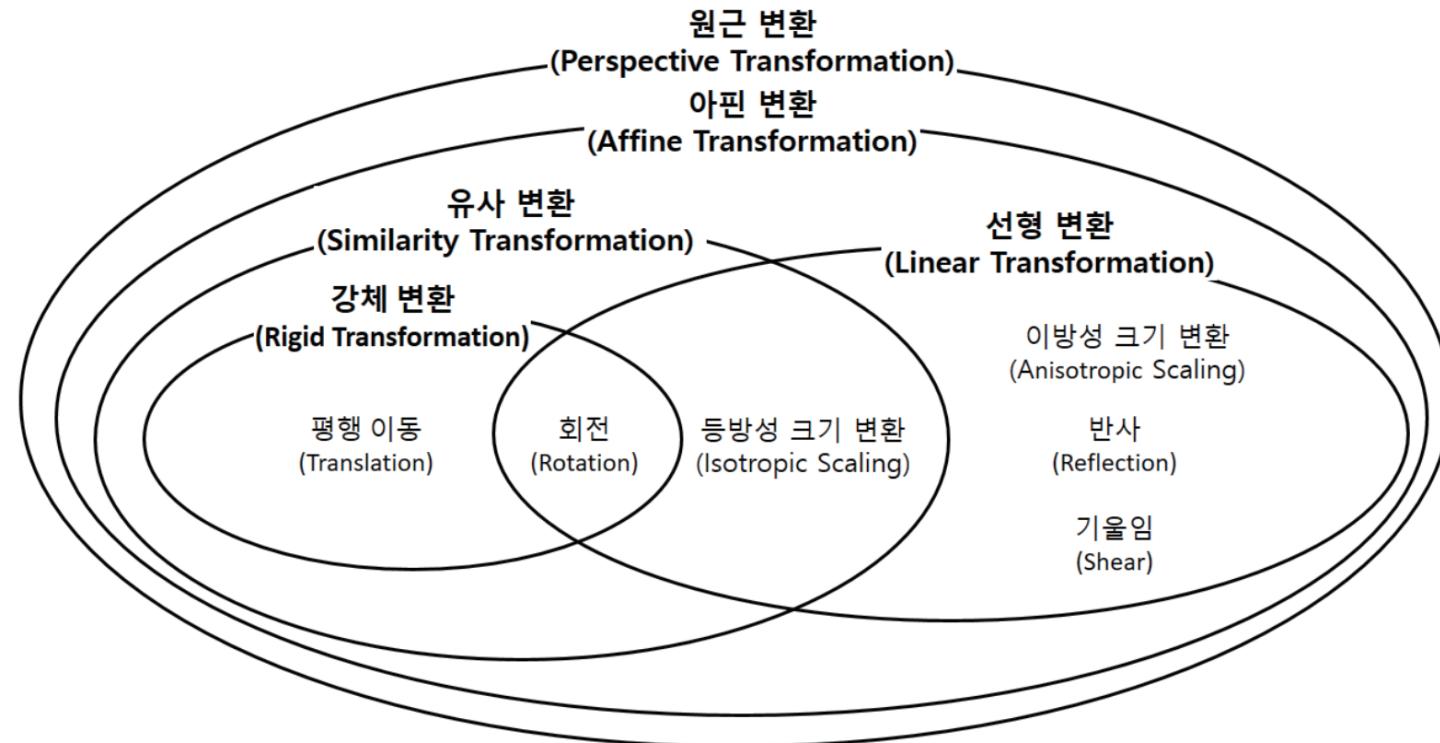


그림 6.1 변환 간의 관계를 집합으로 표현

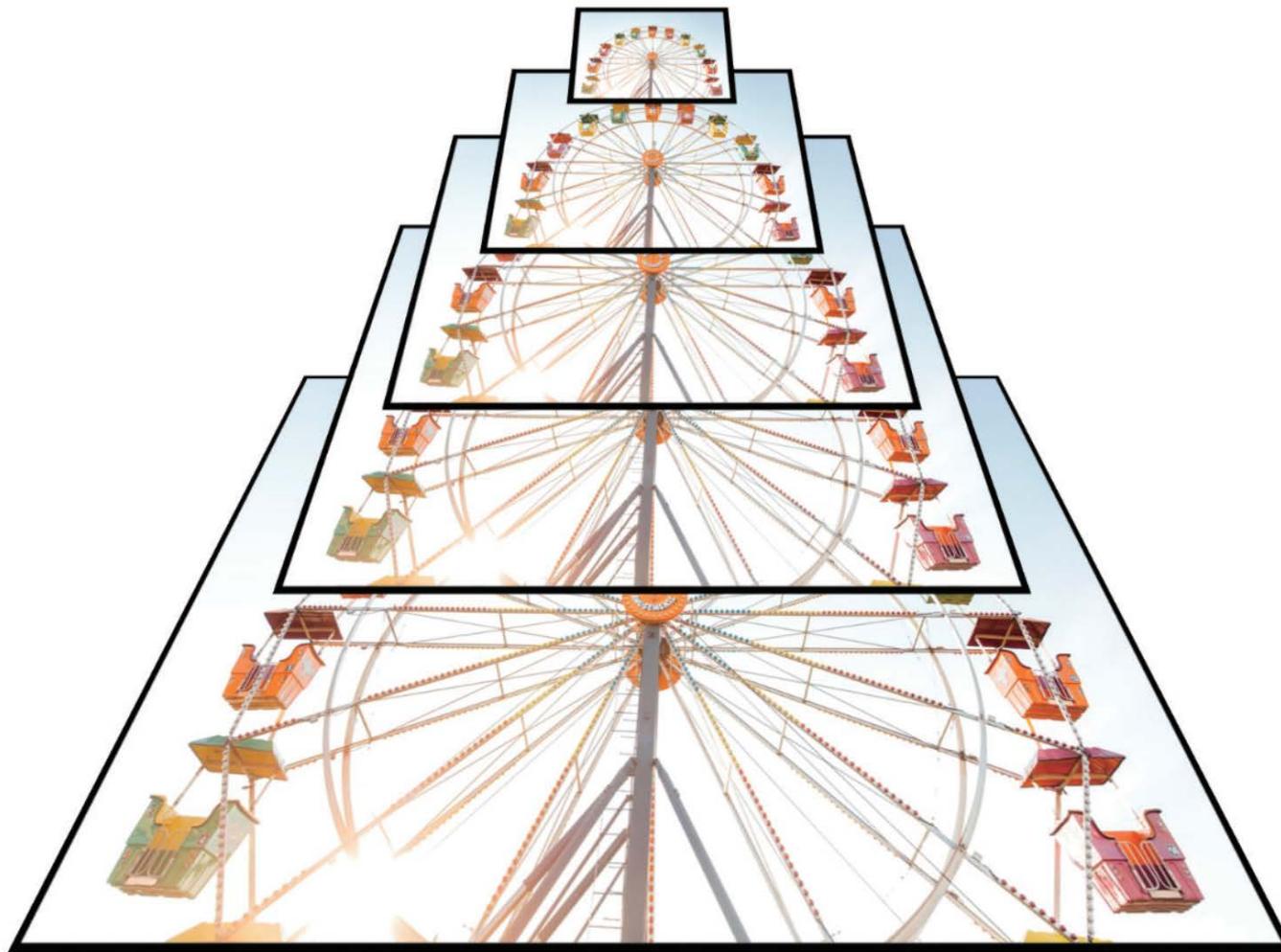


그림 6.2 이미지 피라미드

라플라시안 피라미드 수식:

$$G_0 = \text{Image}$$

$$G_{i+1} = \text{Down}(G_i)$$

$$L_i = G_i - \text{Up}(G_{i+1})$$

출력 이미지 크기 조건:

$$|dstsize.width - src.cols \times 2| \leq (dstsize.width \bmod 2)$$

$$|dstsize.height - src.rows \times 2| \leq (dstsize.height \bmod 2)$$

출력 이미지 크기 조건:

$$|dstsize.width \times 2 - src.cols| \leq 2$$

$$|dstsize.height \times 2 - src.rows| \leq 2$$

$$\frac{1}{256}$$

1	4	6	4	1
4	16	25	16	4
6	24	36	24	6
4	16	25	16	4
1	4	6	4	1

그림 6.3 가우시안 커널

예제 6.1 C# OpenCvSharp에서의 이미지 확대

```
using System;
using OpenCvSharp;

namespace Project
{
    class Program
    {
        static void Main(string[] args)
        {
            Mat src = Cv2.ImRead("ferris-wheel.jpg");
            Mat dst = new Mat(src.Size(), MatType.CV_8UC3);

            Cv2.PyrUp(src, dst, new Size(src.Width*2+1, src.Height*2-1));

            Cv2.ImShow("dst", dst);
            Cv2.WaitKey(0);
            Cv2.DestroyAllWindows();
        }
    }
}
```

【출력 결과】



예제 6.2 Python OpenCV에서의 이미지 축소

```
import cv2

src = cv2.imread("ferris-wheel.jpg")
dst = src.copy()

for i in range(3):
    dst = cv2.pyrDown(dst)

cv2.imshow("dst", dst)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

[출력 결과]



절대 크기(dsize) 변환 수식:

dsize.width = round(fx × src.cols)

dsize.height = round(fy × src.rows)

상대 크기(fx, fy) 변환 수식:

$fx = \text{dsize.width} / \text{src.cols}$

$fy = \text{dsize.height} / \text{src.rows}$

표 6.1 이미지 크기 조절 함수의 보간법

언어	속성	설명
C#	InterpolationFlags.Nearest	가장 가까운 이웃 보간법
Py	cv2.INTER_NEAREST	
C#	InterpolationFlags.Linear	
Py	cv2.INTER_LINEAR	쌍 선형 보간법(파이썬은 비트 단위 쌍 선형 보간법도 지원)
Py	cv2.INTER_LINEAR_EXACT	
C#	InterpolationFlags.Area	영역 보간법
Py	cv2.INTER_AREA	
C#	InterpolationFlags.Cubic	4×4 바이 큐빅 보간법
Py	cv2.INTER_CUBIC	
C#	InterpolationFlags.Lanczos4	8×8 란초스 보간법
Py	cv2.INTER_LANZOS4	

예제 6.3 C# OpenCvSharp에서의 이미지 크기 조절

```
using System;
using OpenCvSharp;

namespace Project
{
    class Program
    {
        static void Main(string[] args)
        {
            Mat src = Cv2.ImRead("car.png");
            Mat dst = new Mat(new Size(1, 1), MatType.CV_8UC3);

            dst = src.SubMat(280, 310, 240, 405);
            Cv2.Resize(dst, dst, new Size(999, 0), 2.0, 2.0, InterpolationFlags.Cubic);

            Cv2.ImShow("dst", dst);
            Cv2.WaitKey(0);
            Cv2.DestroyAllWindows();
        }
    }
}
```

[출력 결과]

D M XG 6271

예제 6.4 Python OpenCV의 이미지 크기 조절

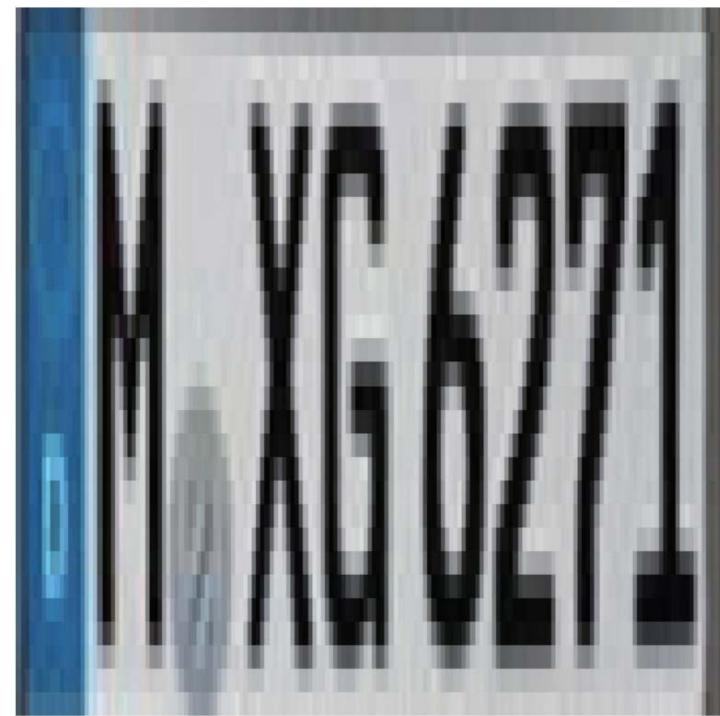
```
import cv2

src = cv2.imread("car.png")

dst = src[280:310, 240:405]
dst = cv2.resize(dst, dsize=(256, 256), interpolation=cv2.INTER_NEAREST)

cv2.imshow("dst", dst)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

【출력 결과】



SRC

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Flip Y

4	3	2	1
8	7	6	5
12	11	10	9
16	15	14	13

13	14	15	16
9	10	11	12
5	6	7	8
1	2	3	4

Flip X

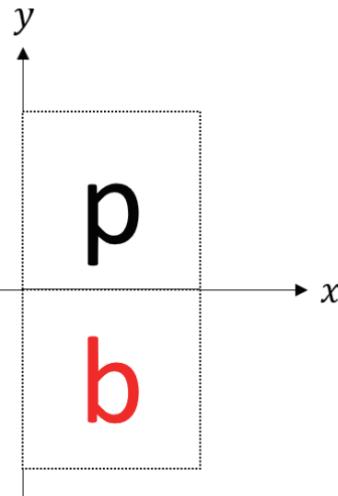
16	15	14	13
12	11	10	9
8	7	6	5
4	3	2	1

Flip XY

그림 6.4 축 대칭 행렬

Flip X

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \times src$$



Flip Y

$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \times src$$

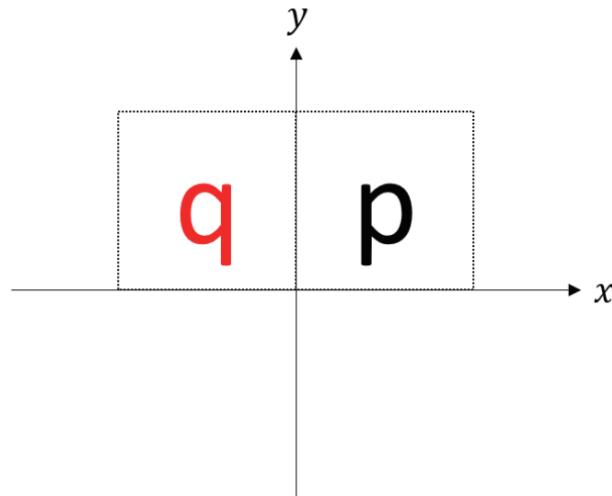


그림 6.5 반사 행렬

표 6.2 대칭 함수 축 플래그

언어	속성	설명
C#	FlipMode.XY	XY축 대칭(상하좌우 대칭)
Py	flipCode < 0	
C#	FlipMode.X	X축 대칭(상하 대칭)
Py	flipCode = 0	
C#	FlipMode.Y	Y축 대칭(좌우 대칭)
Py	flipCode > 0	

좌표의 회전 이동

좌표축의 회전 이동

$$\begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

$$\begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix}$$

그림 6.6 회전 행렬

2×3 회전 행렬

$$\begin{bmatrix} \alpha & \beta & (1 - \alpha) \times Center_x - \beta \times Center_y \\ -\beta & \alpha & \beta \times Center_x - (1 - \alpha) \times Center_y \\ \alpha = scale \times \cos \theta \\ \beta = scale \times \sin \theta \end{bmatrix}$$

그림 6.7 2×3 회전 행렬

예제 6.5 C# OpenCvSharp에서의 축 대칭과 회전

```
using System;
using OpenCvSharp;

namespace Project
{
    class Program
    {
        static void Main(string[] args)
        {
            Mat src = Cv2.ImRead("glass.jpg");
            Mat dst = new Mat();

            Cv2.Flip(src, dst, FlipMode.Y);
            Mat matrix = Cv2.GetRotationMatrix2D(new Point2f(src.Width / 2, src.Height / 2), 90.0,
1.0);
            Cv2.WarpAffine(dst, dst, matrix, new Size(src.Width, src.Height));

            Cv2.ImShow("dst", dst);
            Cv2.WaitKey(0);
            Cv2.DestroyAllWindows();
        }
    }
}
```

[출력 결과]



예제 6.6 Python OpenCV에서의 회전 행렬의 재할당

```
import math
import cv2

src = cv2.imread("glass.jpg")

height, width, _ = src.shape
center = (width / 2, height / 2)
angle = 90
scale = 0.5

matrix = cv2.getRotationMatrix2D(center, angle, scale)

radians = math.radians(angle)
sin = math.sin(radians)
cos = math.cos(radians)
bound_w = int((height * scale * abs(sin)) + (width * scale * abs(cos)))
bound_h = int((height * scale * abs(cos)) + (width * scale * abs(sin)))

matrix[0, 2] += ((bound_w / 2) - center[0])
matrix[1, 2] += ((bound_h / 2) - center[1])

dst = cv2.warpAffine(src, matrix, (bound_w, bound_h))

cv2.imshow("dst", dst)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

[출력 결과]



회전 후의 이미지 크기:

$$\text{boundingWidth} = \text{height} \times \text{scale} \times \sin \theta + \text{width} \times \text{scale} \times \cos \theta$$

$$\text{boundingHeight} = \text{height} \times \text{scale} \times \cos \theta + \text{width} \times \text{scale} \times \sin \theta$$

회전 후의 이미지 중심점:

$$\text{matrix}[0, 2] += \text{boundingCenter}_x - \text{Center}_x$$

$$\text{matrix}[1, 2] += \text{boundingCenter}_y - \text{Center}_y$$

$$\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} a_{00} & a_{01} & b_0 \\ a_{10} & a_{11} & b_1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} a_{00}x_1 + a_{01}y_1 + b_0 \\ a_{10}x_1 + a_{11}y_1 + b_1 \\ 0 + 0 + 1 \end{bmatrix}$$

그림 6.8 아핀 변환 행렬

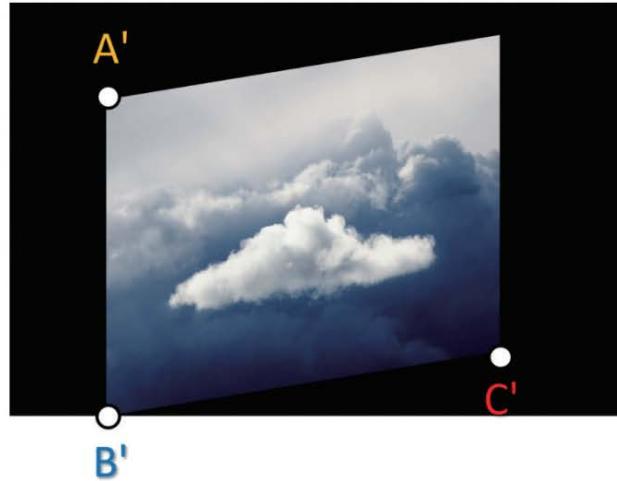
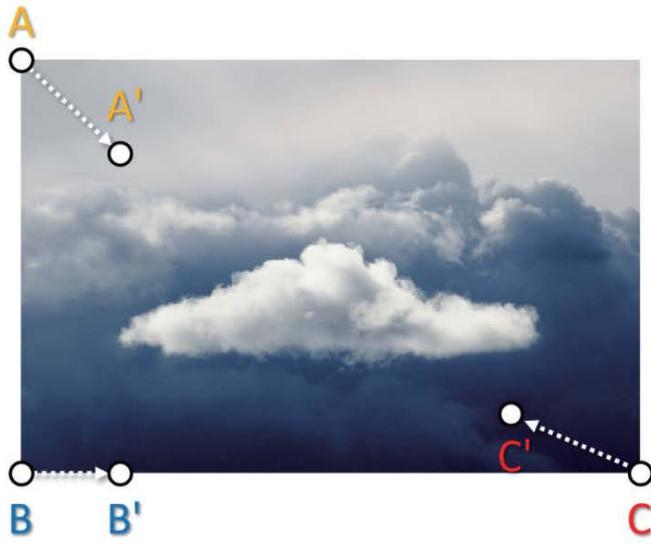


그림 6.9 3점 매핑 아핀 변환

$$\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} a_{00} & a_{01} & b_0 \\ a_{10} & a_{11} & b_1 \\ a_{20} & a_{21} & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} a_{00}x_1 + a_{01}y_1 + b_0 \\ a_{10}x_1 + a_{11}y_1 + b_1 \\ a_{20}x_1 + a_{21}y_1 + 1 \end{bmatrix}$$

그림 6.10 원근 변환 행렬

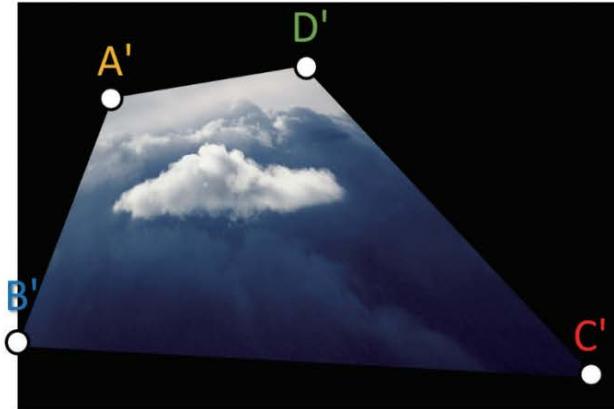
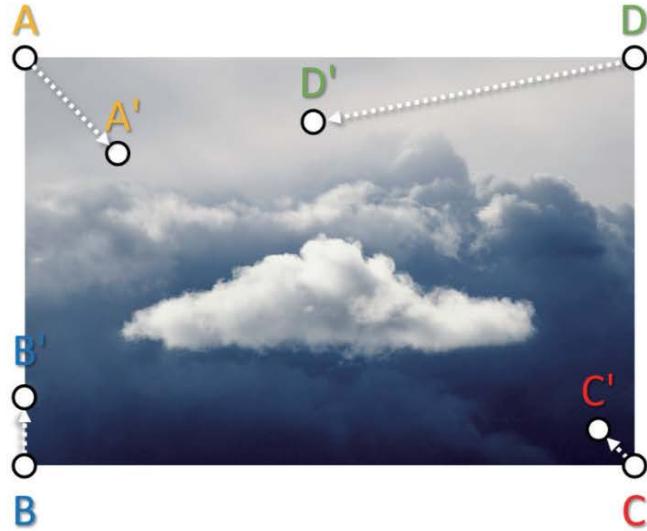


그림 6.11 4점 매핑 원근 변환

예제 6.7 C# OpenCvSharp에서의 아핀 변환

```
using System;
using System.Collections.Generic;
using OpenCvSharp;

namespace Project
{
    class Program
    {
        static void Main(string[] args)
        {
            Mat src = Cv2.ImRead("clouds.jpg");
            Mat dst = new Mat();

            List<Point2f> src_pts = new List<Point2f>()
            {
                new Point2f(0.0f, 0.0f),
                new Point2f(0.0f, src.Height),
                new Point2f(src.Width, src.Height)
            };

            List<Point2f> dst_pts = new List<Point2f>()
            {
                new Point2f(300.0f, 300.0f),
                new Point2f(300.0f, src.Height),
                new Point2f(src.Width-400.0f, src.Height-200.0f)
            };
        }
    }
}
```

```
Mat M = Cv2.GetAffineTransform(src_pts, dst_pts);

Cv2.WarpAffine(
    src, dst, M, new Size(src.Width, src.Height),
    borderValue: new Scalar(127, 127, 127, 0)
);

Cv2.ImShow("dst", dst);
Cv2.WaitKey(0);
Cv2.DestroyAllWindows();

}
}
}
```

[출력 결과]



예제 6.8 Python OpenCV에서의 원근 변환

```
import numpy as np
import cv2

src = cv2.imread("clouds.jpg")
height, width, _ = src.shape

src_pts = np.array([[0.0, 0.0], [width, 0.0], [width, height], [0.0, height]], dtype=np.float32)
dst_pts = np.array([[300.0, 300.0], [900.0, 200.0], [width-100.0, height-100.0], [0.0, height-200.0]], dtype=np.float32)

M = cv2.getPerspectiveTransform(src_pts, dst_pts)

dst = cv2.warpPerspective(src, M, (width, height), borderValue=(255, 255, 255, 0))

cv2.imshow("dst", dst)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

[출력 결과]



팽창과 침식 수식:

$$dilate(x, y) = \max_{(i, j) \in kernel} src(x + i, y + j)$$

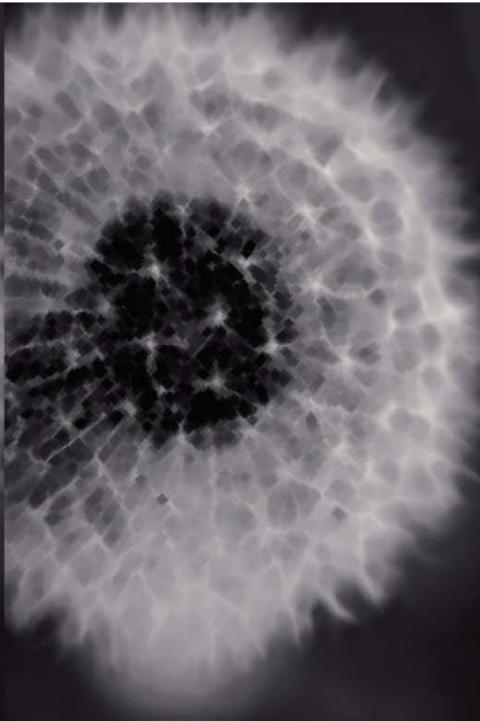
$$erode(x, y) = \min_{(i, j) \in kernel} src(x + i, y + j)$$



(a) 원본 이미지

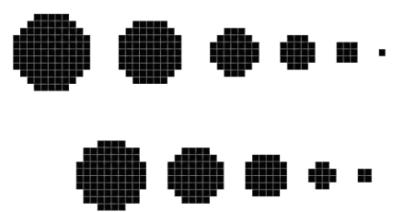


(b) 팽창(dilation)

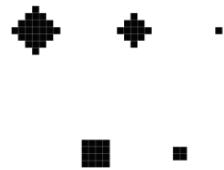


(c) 침식(erosion)

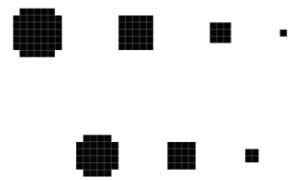
그림 6.12 모폴로지 변환



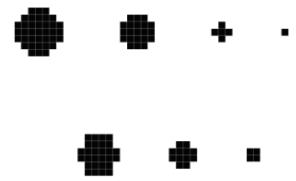
(a) 원본 이미지



(b) 직사각형 구조 요소

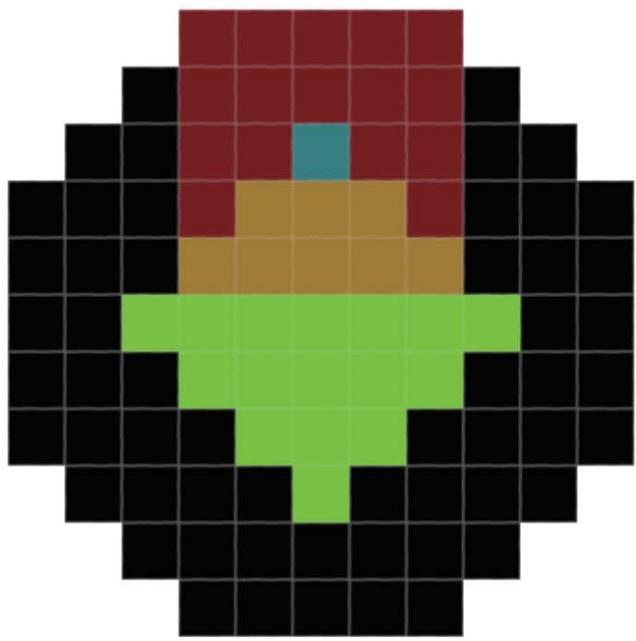


(c) 십자가 구조 요소

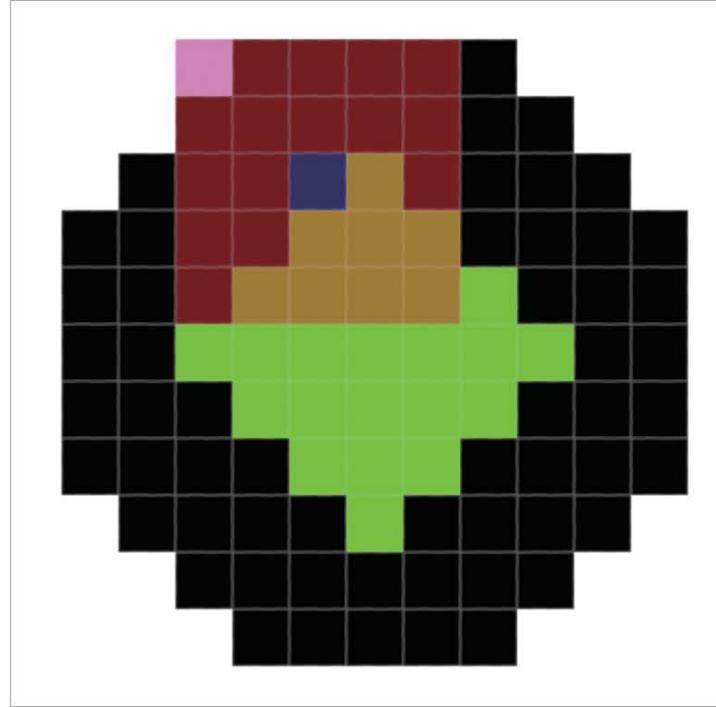


(d) 타원 구조 요소

그림 6.13 직사각형, 십자가, 타원 구조 요소를 적용한 팽창 연산



(a) 제거되는 영역



(b) 유지되는 영역

그림 6.14 구조 요소에 따른 픽셀의 변화

표 6.3 구조 요소 생성 함수 형태 플래그

언어	속성	설명
C#	MorphShapes.Rect	직사각형 형태. 모든 i, j의 요소에 대해 $K_{i,j}=1$
Py	cv2.MORPH_RECT	
C#	MorphShapes.Cross	십자가 형태. 고정점의 i, j에 대해 $K_{i,j}=1$
Py	cv2.MORPH_CROSS	
C#	MorphShapes.Ellipse	타원 형태. 커널의 너비와 높이를 축으로 하는 타원에 대해 $K_{i,j}=1$
Py	cv2.MORPH_ELLIPSE	

예제 6.9 C# OpenCvSharp에서의 모폴로지 팽창

```
using System;
using OpenCvSharp;

namespace Project
{
    class Program
    {
        static void Main(string[] args)
        {
            Mat src = Cv2.ImRead("dandelion.jpg", ImreadModes.Grayscale);
            Mat dst = new Mat();

            Mat kernel= Cv2.GetStructuringElement(MorphShapes.Cross, new Size(7, 7));
            Cv2.Dilate(src, dst, kernel, new Point(-1, -1), 3, BorderTypesReflect101, new
Scalar(0));

            Cv2.ImShow("dst", dst);
            Cv2.WaitKey(0);
            Cv2.DestroyAllWindows();
        }
    }
}
```

[출력 결과]



예제 6.10 Python OpenCV의 모폴로지 침식

```
import cv2

src = cv2.imread("dandelion.jpg", cv2.IMREAD_GRAYSCALE)

kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (5, 5), anchor=(-1, -1))
dst = cv2.erode(src, kernel, iterations=3)

cv2.imshow("dst", dst)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

[출력 결과]

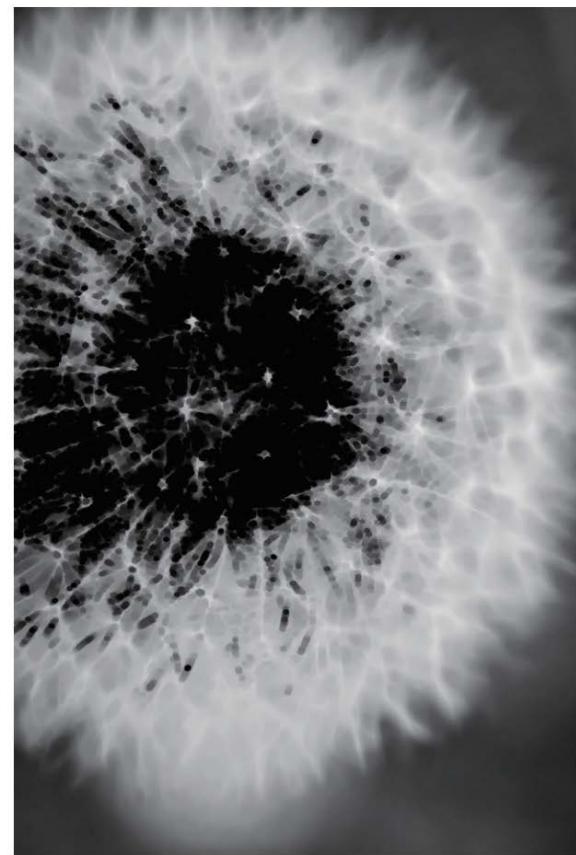


표 6.4 모폴로지 연산 함수의 연산자 플래그

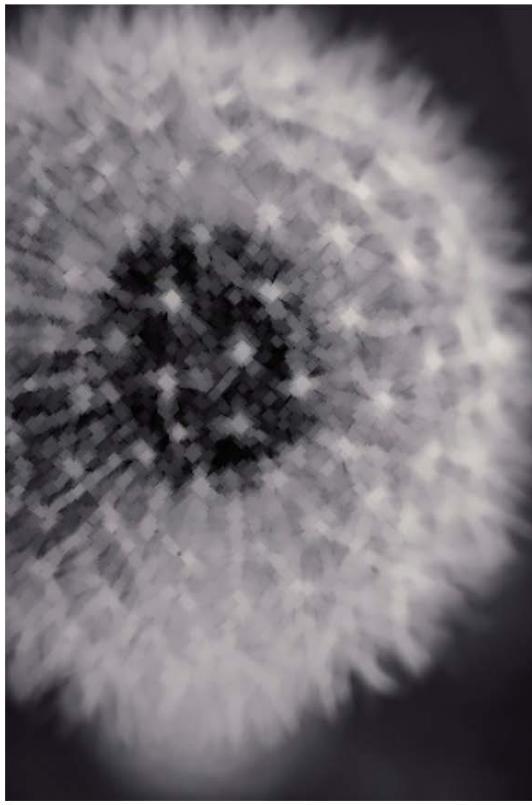
언어	속성	설명
C#	MorphTypes.Dilate	팽창 연산
Py	cv2.MORPH_DILATE	
C#	MorphTypes.Erode	침식 연산
Py	cv2.MORPH_ERODE	
C#	MorphTypes.Open	열림 연산
Py	cv2.MORPH_OPEN	
C#	MorphTypes.Close	닫힘 연산
Py	cv2.MORPH_CLOSE	
C#	MorphTypes.Gradient	모폴로지 그레이디언트
Py	cv2.MORPH_GRADIENT	
C#	MorphTypes.TopHat	탑햇 연산
Py	cv2.MORPH_TOPHAT	
C#	MorphTypes.BlackHat	블랙햇 연산
Py	cv2.MORPH_BLACKHAT	
C#	MorphTypes.HitMiss	히트미스 연산
Py	cv2.MORPH_HITMISS	

열림 연산:

```
dst = dilate(erode(src))
```



(a) 원본 이미지



(b) 열림 연산

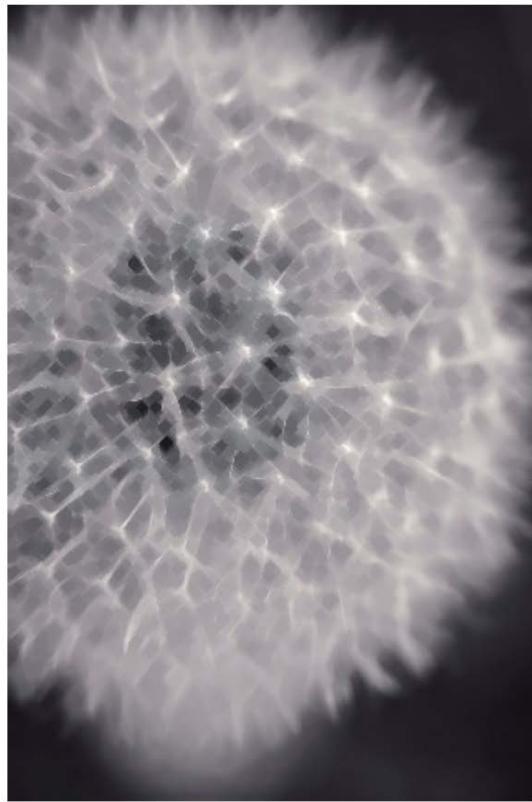
그림 6.15 모폴로지 열림 연산을 적용한 이미지

닫힘 연산:

```
dst = erode(dilate(src))
```



(a) 원본 이미지



(b) 닫힘 연산

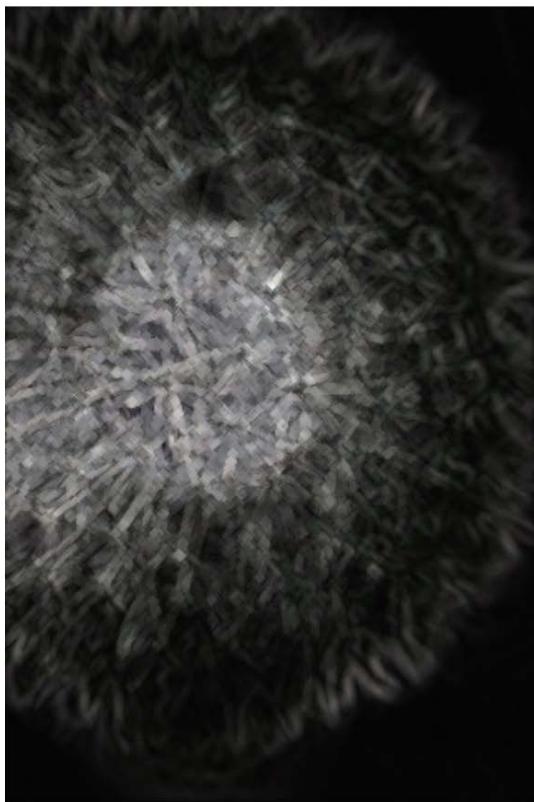
그림 6.16 모폴로지 닫힘 연산을 적용한 이미지

그레이디언트 연산:

$$\text{dst} = \text{dilate}(\text{src}) - \text{erode}(\text{src})$$



(a) 원본 이미지



(b) 그레이디언트 연산

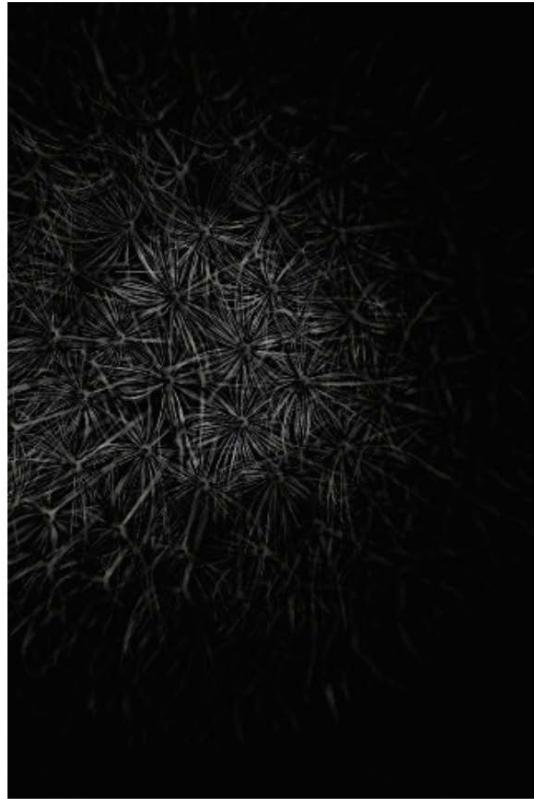
그림 6.17 모폴로지 그레이디언트 연산을 적용한 이미지

탭햇 연산:

$\text{dst} = \text{src} - \text{open}(\text{src})$



(a) 원본 이미지



(b) 탑햇 연산

그림 6.18 모폴로지 탑햇 연산을 적용한 이미지

블랙햇 연산:

$\text{dst} = \text{close}(\text{src}) - \text{src}$



(a) 원본 이미지



(b) 블랙햇 연산

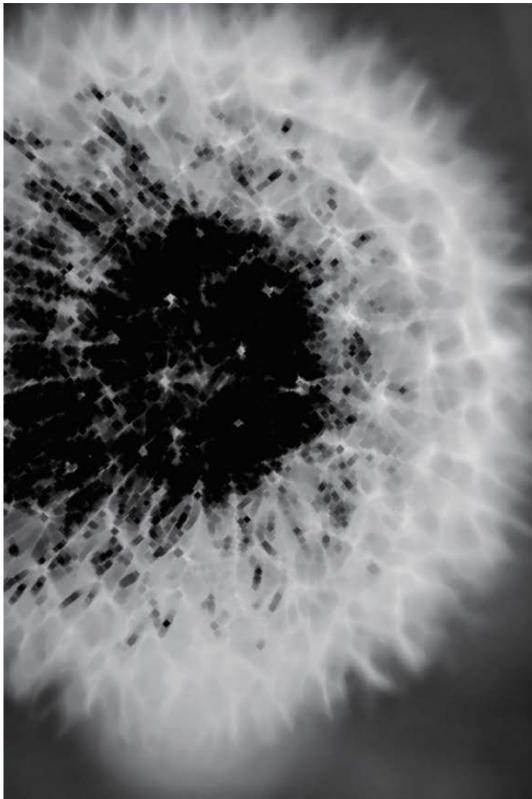
그림 6.19 모폴로지 블랙햇 연산을 적용한 이미지

히트미스 제한 조건:

8-bit unsigned integers, 1-Channel



(a) 원본 이미지



(b) 히트미스 연산

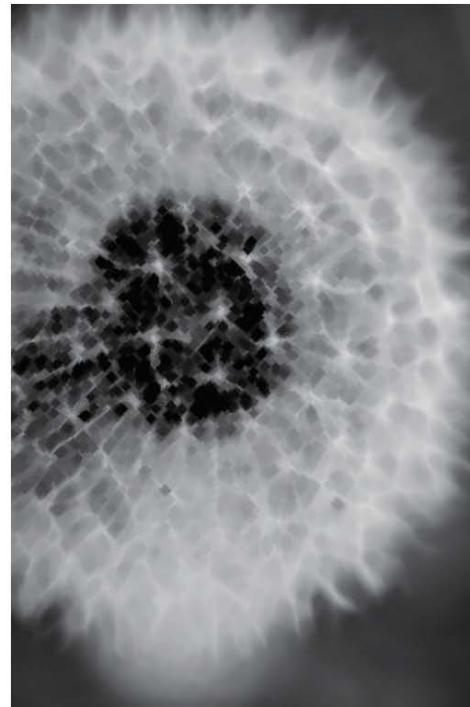
그림 6.20 모폴로지 히트미스 연산을 적용한 이미지



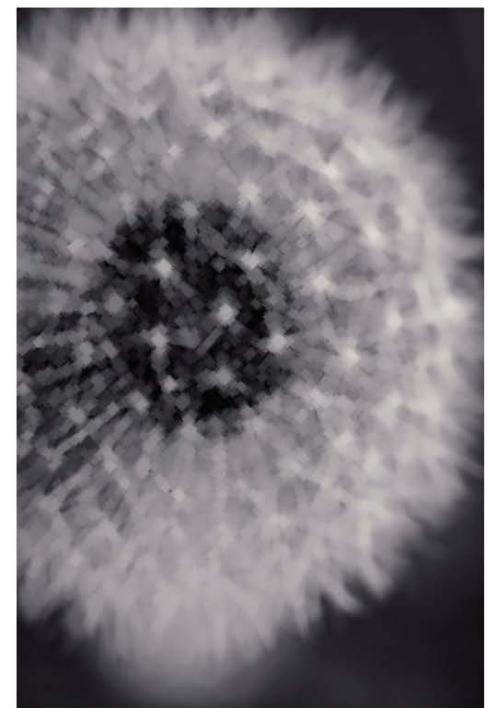
(a) 원본 이미지



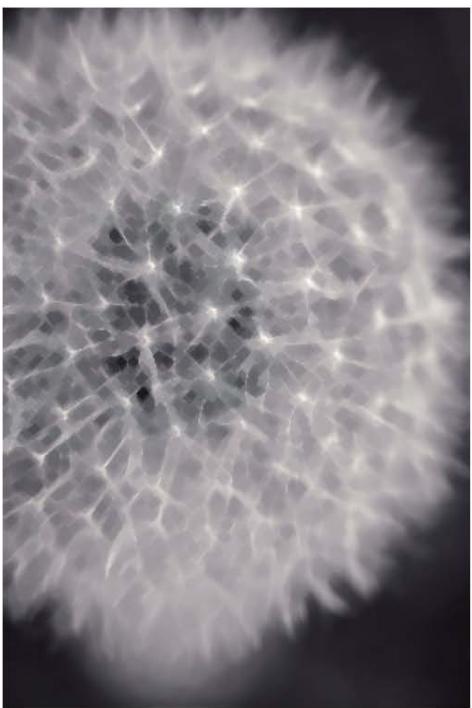
(b) 팽창 연산



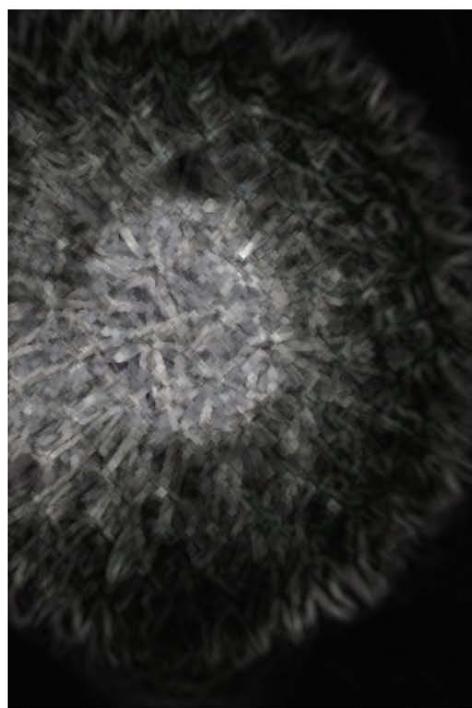
(c) 침식 연산



(d) 열림 연산



(e) 닫힘 연산



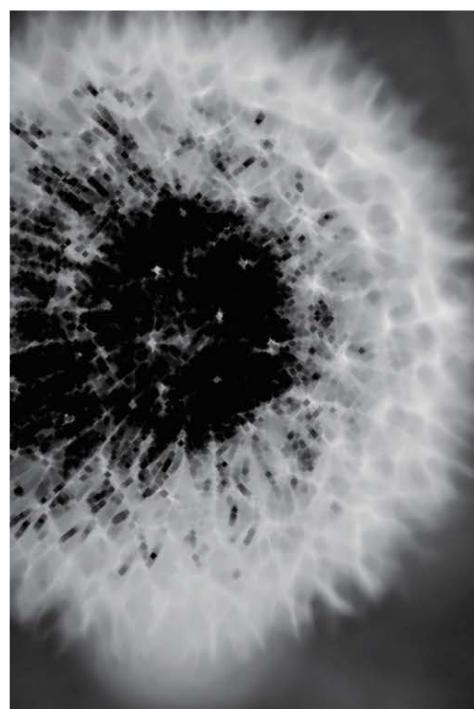
(f) 그레이디언트 연산



(g) 탑햇 연산



(h) 블랙햇 연산



(i) 히트미스 연산

그림 6.21 모풀로지 연산의 형태

예제 6.11 C# OpenCvSharp 모풀로지 히트미스

```
using System;
using OpenCvSharp;

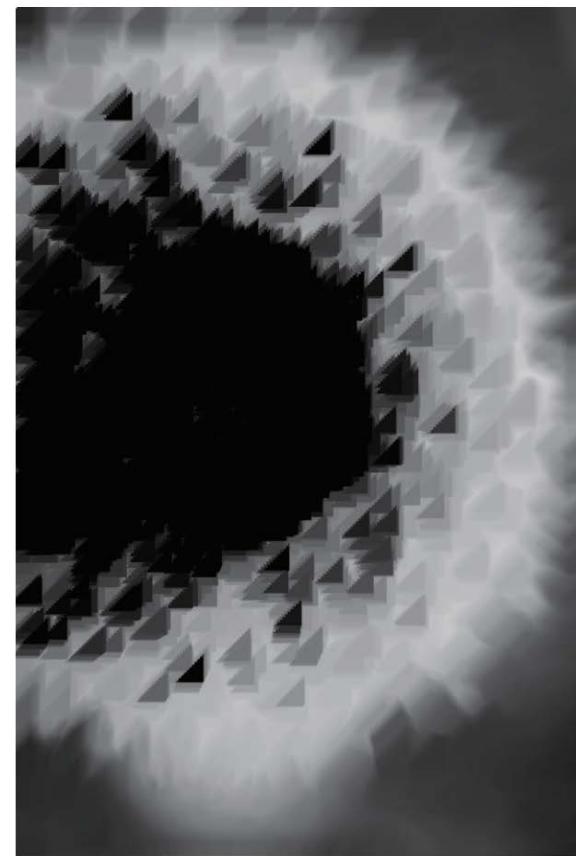
namespace Project
{
    class Program
    {
        static void Main(string[] args)
        {
            Mat src = Cv2.ImRead("dandelion.jpg", ImreadModes.Grayscale);
            Mat dst = new Mat();

            Mat kernel = Mat.Zeros(new Size(7, 7), MatType.CV_8UC1);
            kernel[0, 7, 0, 1] = Mat.Ones(new Size(1, 7), MatType.CV_8UC1);
            kernel[0, 1, 0, 7] = Mat.Ones(new Size(7, 1), MatType.CV_8UC1);

            Cv2.MorphologyEx(src, dst, MorphTypes.HitMiss, kernel, iterations: 10);

            Cv2.ImShow("dst", dst);
            Cv2.WaitKey(0);
            Cv2.DestroyAllWindows();
        }
    }
}
```

[출력 결과]



예제 6.12 Python OpenCV에서의 모폴로지 히트미스

```
import numpy as np
import cv2

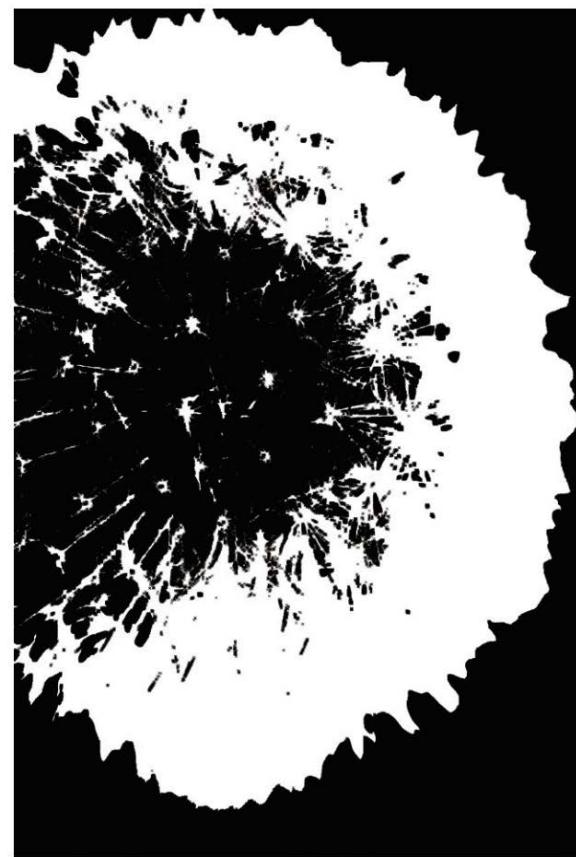
src = cv2.imread("dandelion.jpg", cv2.IMREAD_GRAYSCALE)
_, binary = cv2.threshold(src, 127, 255, cv2.THRESH_BINARY)

kernel = np.array([[1, 0, 0, 0, 1],
                  [0, 1, 0, 1, 0],
                  [0, 0, 1, 0, 0],
                  [0, 1, 0, 1, 0],
                  [1, 0, 0, 0, 1]])

dst = cv2.morphologyEx(binary, cv2.MORPH_HITMISS, kernel, iterations=1)

cv2.imshow("dst", dst)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

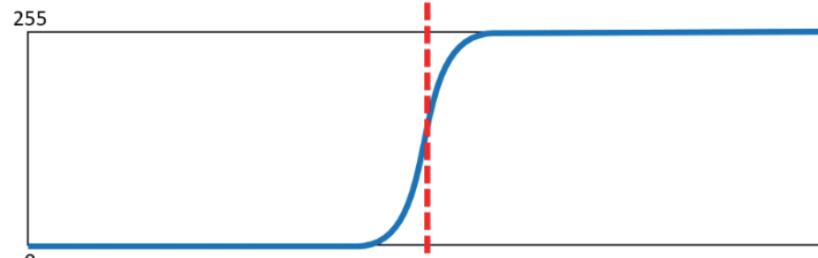
[출력 결과]



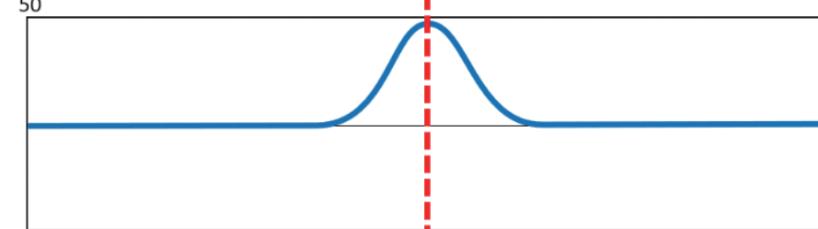
입력 이미지



픽셀의 밝기
그래프



1차 미분
그래프



2차 미분
그래프

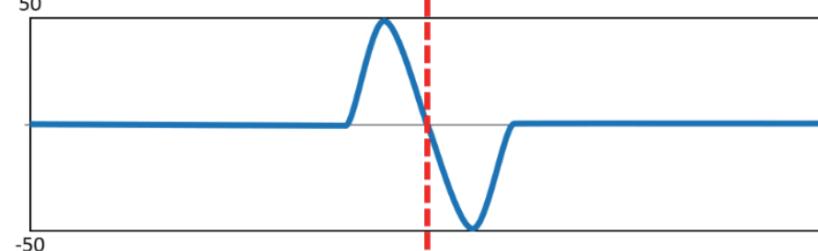


그림 7.1 가장자리의 밝기 그래프



(a) Step Edge



(b) Line Edge



(c) Ramp Edge



(d) Roof Edge

그림 7.2 가장자리 유형

-1	0	1
-2	0	1
-1	0	1

(a) 3×3 수직 마스크

1	2	1
0	0	0
-1	-2	-1

(b) 3×3 수평 마스크

-1	-1	0	1	1
-1	-1	0	1	1
-2	-2	0	2	2
-1	-1	0	1	1
-1	-1	0	1	1

(c) 5×5 수직 마스크

1	1	2	1	1
1	1	2	1	1
0	0	0	0	0
-1	-1	-2	-1	-1
-1	-1	-2	-1	-1

(d) 5×5 수평 마스크

그림 7.3 소벨 마스크 형태

소벨 연산 형태:

$$\text{dst} = \frac{\partial}{\partial x^{xorder}} \frac{\partial}{\partial y^{yorder}} \text{src}$$

-3	0	3
-10	0	10
-3	0	3

(a) 3×3 수직 마스크

-3	-10	-3
0	0	0
3	10	3

(b) 3×3 수평 마스크

그림 7.4 샤르 필터 형태

라플라시안 연산:

$$\text{Laplace}(f) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

0	1	0
1	-4	1
0	1	0

-1	-1	-1
-1	8	-1
-1	-1	-1

(a) 라플라시안 단일 커널

(b) 또 다른 형태의 라플라시안 커널

그림 7.5 라플라시안 필터 형태

리플라시안 연산 형태:

$$\text{dst} = \Delta \text{src} = \frac{\partial^2 \text{src}}{\partial x^2} + \frac{\partial^2 \text{src}}{\partial y^2}$$

L_1 -norm 수식:

$$L_1 = \left| \frac{dI}{dx} \right| + \left| \frac{dI}{dy} \right|$$

L_2 -norm 수식:

$$L_2 = \sqrt{\left(\frac{dI}{dx}\right)^2 + \left(\frac{dI}{dy}\right)^2}$$

예제 7.1 C# OpenCvSharp에서의 소벨 연산

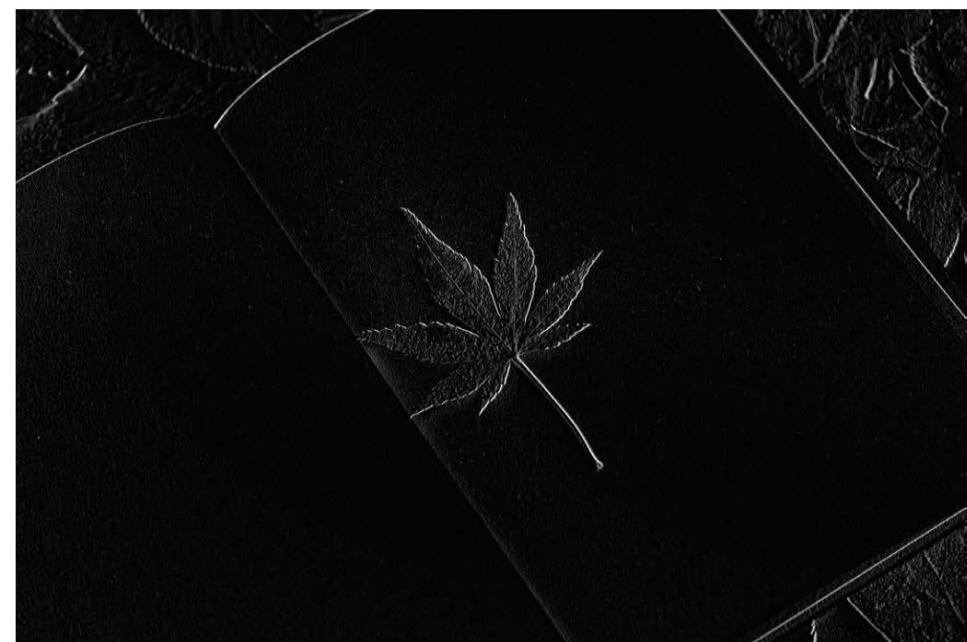
```
using System;
using OpenCvSharp;

namespace Project
{
    class Program
    {
        static void Main(string[] args)
        {
            Mat src = Cv2.ImRead("book.jpg", ImreadModes.Grayscale);
            Mat dst = new Mat();

            Cv2.Sobel(src, dst, MatType.CV_8UC1, 1, 0, 3, 1, 0, BorderTypesReflect101);

            Cv2.ImShow("dst", dst);
            Cv2.WaitKey(0);
            Cv2.DestroyAllWindows();
        }
    }
}
```

[출력 결과]



예제 7.2 Python OpenCV에서의 캐니 엣지

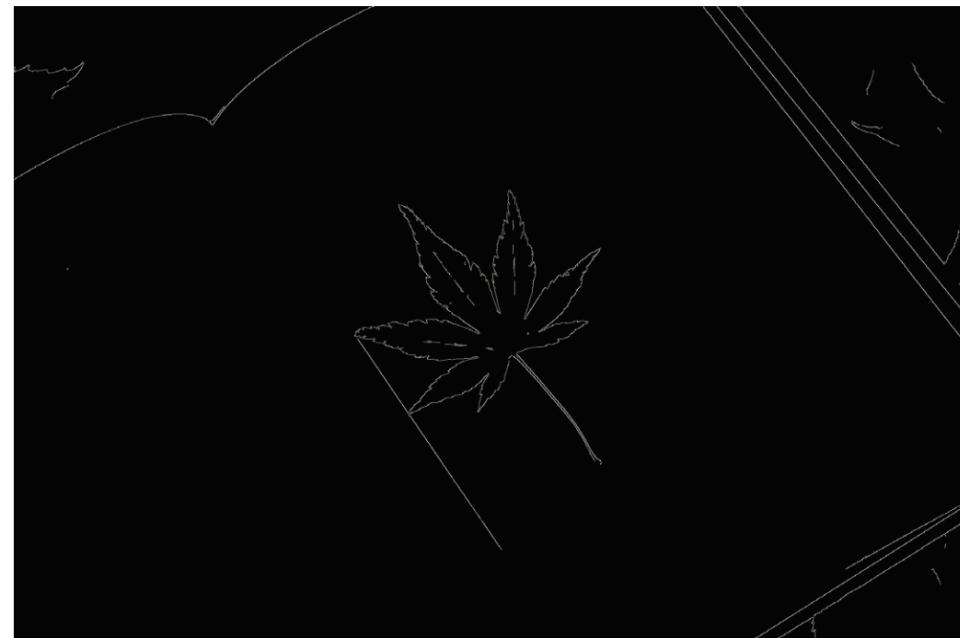
```
import numpy as np
import cv2

src = cv2.imread("book.jpg", cv2.IMREAD_GRAYSCALE)

dst = cv2.Canny(src, 100, 200, apertureSize=3, L2gradient=True)

cv2.imshow("dst", dst)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

[출력 결과]



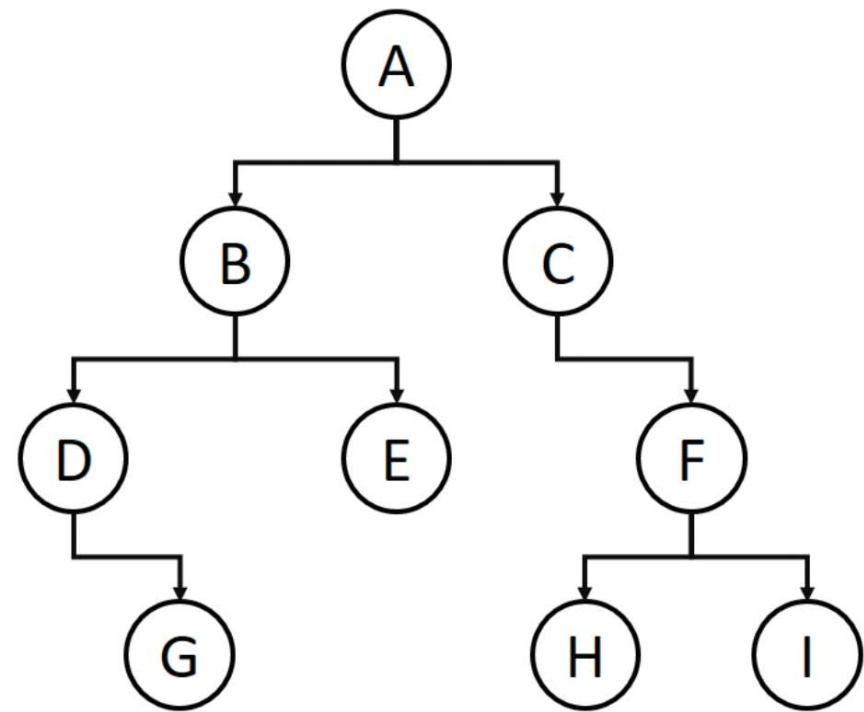
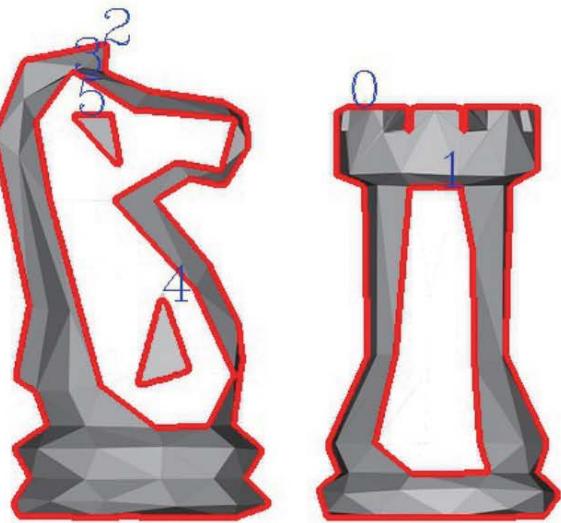


그림 7.6 트리 구조의 형태



노드	다음 윤곽선	이전 윤곽선	자식 노드	부모 노드
0	2	-1	1	-1
1	-1	-1	-1	0
2	-1	0	3	-1
3	-1	-1	4	2
4	5	-1	-1	3
5	-1	4	-1	3

그림 7.7 계층 구조의 형태

표 7.1 검색 방법 플래그

언어	속성	설명
C#	RetrievalModes.External	최외곽 윤곽선만 검색
Py	cv2.RETR_EXTERNAL	
C#	RetrievalModes.List	모든 윤곽선을 검출하며, 계층 구조를 형성하지 않음(모든 윤곽선을 동일 레벨로 간주)
Py	cv2.RETR_LIST	
C#	RetrievalModes.CComp	모든 윤곽선을 검색해서 2단계 계층 구조로 구성(최상위 레벨은 외곽, 두 번째 레벨은 내곽(홀))
Py	cv2.RETR_CCOMP	
C#	RetrievalModes.Tree	모든 윤곽선을 검출하고 트리 구조로 구성
Py	cv2.RETR_TREE	

표 7.2 근사 방법 플래그

언어	속성	설명
C#	ContourApproximationModes.ApproxNone	검출된 모든 윤곽점을 반환
Py	cv2.CHAIN_APPROX_NONE	
C#	ContourApproximationModes.ApproxSimple	수평, 수직, 대각선 부분을 압축해서 끝점만 반환
Py	cv2.CHAIN_APPROX_SIMPLE	
C#	ContourApproximationModes.ApproxTC89L1	
Py	cv2.CHAIN_APPROX_TC89_L1	The-Chin 체인 근사 알고리즘을 적용 ⁸
C#	ContourApproximationModes.ApproxTC89KC0S	
Py	cv2.CHAIN_APPROX_TC89_KCOS	

예제 7.3 C# OpenCvSharp에서의 윤곽선 검출

```
using System;
using OpenCvSharp;

namespace Project
{
    class Program
    {
        static void Main(string[] args)
        {
            Mat src = Cv2.ImRead("chess.png");
            Mat gray = new Mat();
            Mat binary = new Mat();
            Mat morph = new Mat();
            Mat image = new Mat();
            Mat dst = src.Clone();

            Mat kernel = Cv2.GetStructuringElement(MorphShapes.Rect, new Size(3, 3));
            Point[][] contours;
            HierarchyIndex[] hierarchy;

            Cv2.CvtColor(src, gray, ColorConversionCodes.BGR2GRAY);
            Cv2.Threshold(gray, binary, 230, 255, ThresholdTypes.Binary);
            Cv2.MorphologyEx(binary, morph, MorphTypes.Close, kernel, new Point(-1, -1), 2);
            Cv2.BitwiseNot(morph, image);
```

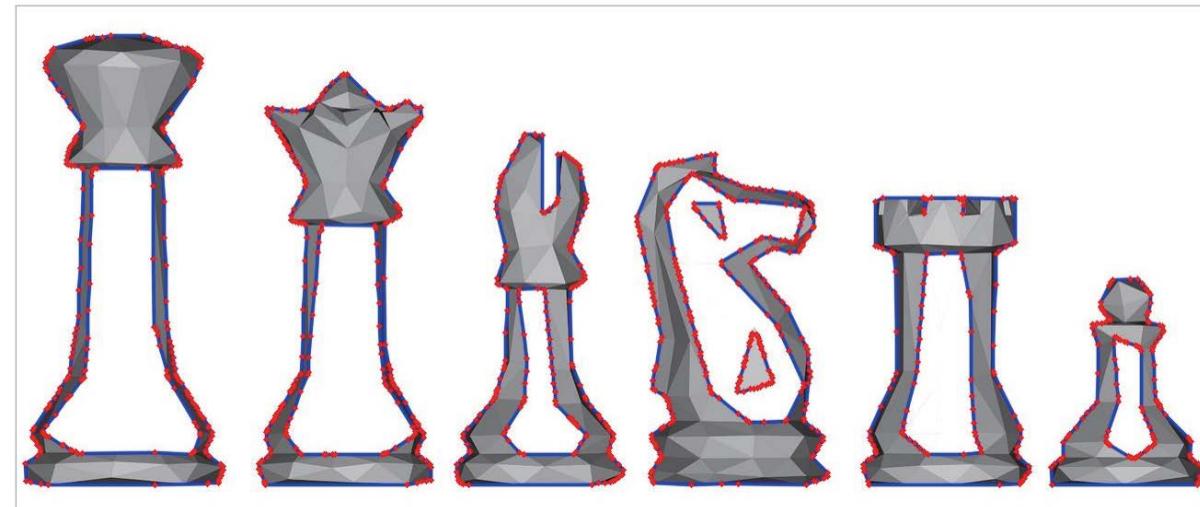
```
Cv2.FindContours(image, out contours, out hierarchy, RetrievalModes.Tree,
ContourApproximationModes.ApproxTC89KCOS);

Cv2.DrawContours(dst, contours, -1, new Scalar(255, 0, 0), 2, LineTypes.AntiAlias,
hierarchy, 3);

for (int i = 0; i < contours.Length; i++)
{
    for (int j = 0; j < contours[i].Length; j++)
    {
        Cv2.Circle(dst, contours[i][j], 1, new Scalar(0, 0, 255), 3);
    }
}

Cv2.ImShow("dst", dst);
Cv2.WaitKey(0);
Cv2.DestroyAllWindows();
}
```

[출력 결과]



예제 7.4 Python OpenCV에서의 윤곽선 검출

```
import cv2

src = cv2.imread("chess.png")
dst = src.copy()

kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (3, 3))

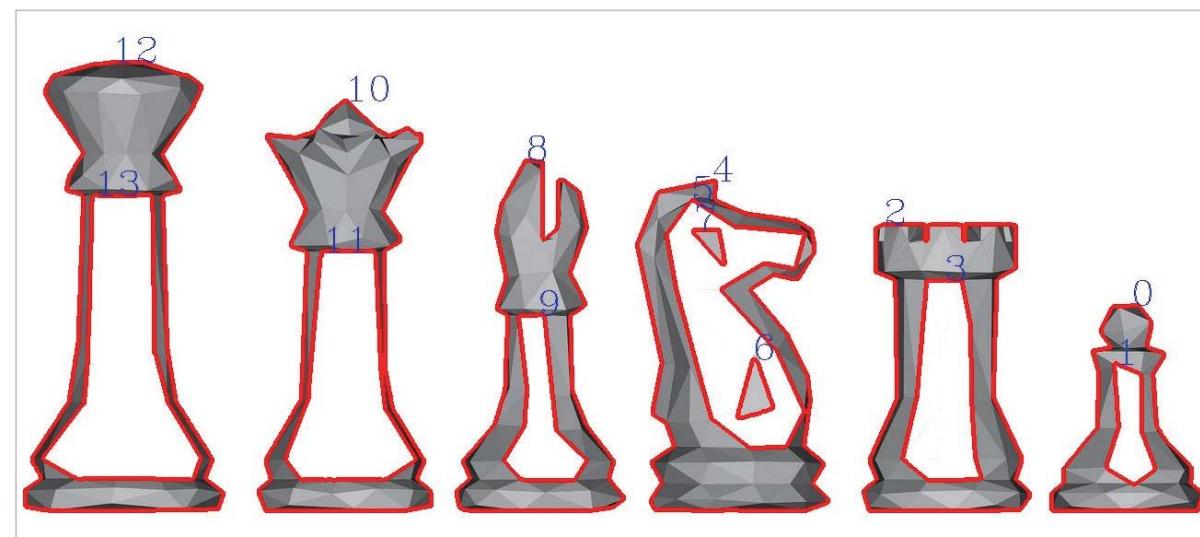
gray = cv2.cvtColor(src, cv2.COLOR_RGB2GRAY)
ret, binary = cv2.threshold(gray, 230, 255, cv2.THRESH_BINARY)
morp = cv2.morphologyEx(binary, cv2.MORPH_CLOSE, kernel, iterations=2)
image = cv2.bitwise_not(morp)

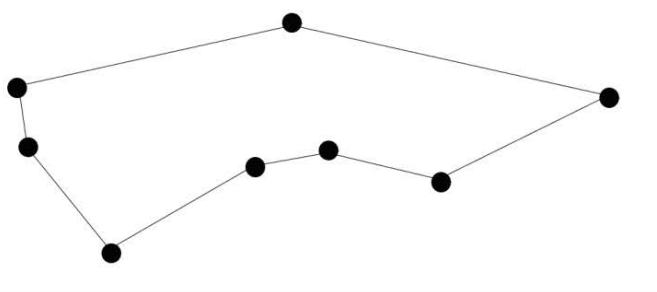
contours, hierarchy = cv2.findContours(image, cv2.RETR_TREE, cv2.CHAIN_APPROX_NONE)
cv2.drawContours(dst, contours, -1, (0, 0, 255), 3)
for i in range(len(contours)):
    cv2.putText(dst, str(i), tuple(contours[i][0][0]), cv2.FONT_HERSHEY_COMPLEX, 1.3, (255, 0, 0), 1)
    print(i, hierarchy[0][i])

cv2.imshow("dst", dst)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

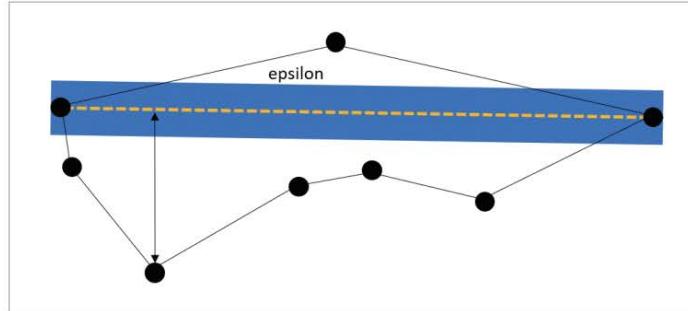
【출력 결과】

```
0 [ 2 -1  1 -1 ]  
1 [-1 -1 -1  0]  
2 [ 4  0  3 -1 ]  
3 [-1 -1 -1  2 ]  
4 [ 8  2  5 -1 ]  
5 [-1 -1  6  4 ]  
6 [ 7 -1 -1  5 ]  
7 [-1  6 -1  5 ]  
8 [10  4  9 -1 ]  
9 [-1 -1 -1  8 ]  
10 [12  8 11 -1 ]  
11 [-1 -1 -1 10 ]  
12 [-1 10 13 -1 ]  
13 [-1 -1 -1 12 ]
```

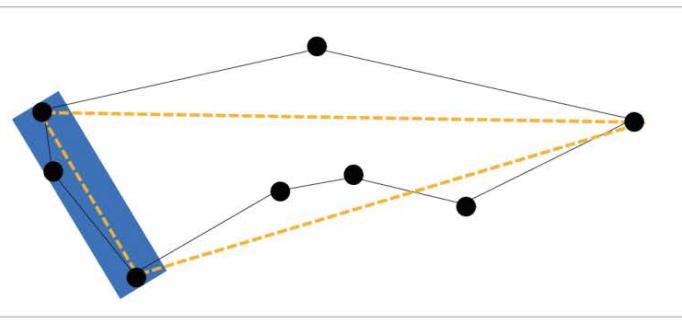




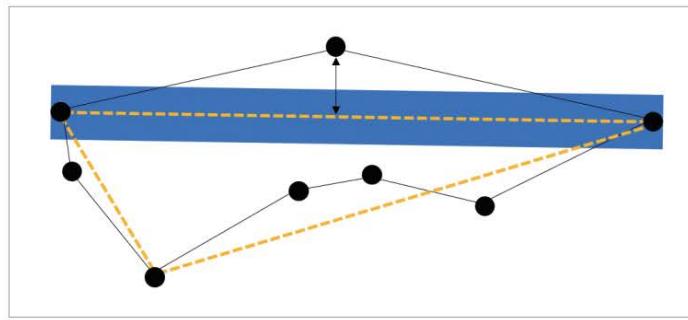
(a)



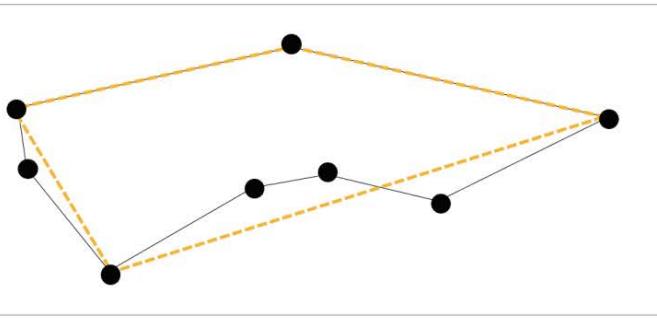
(b)



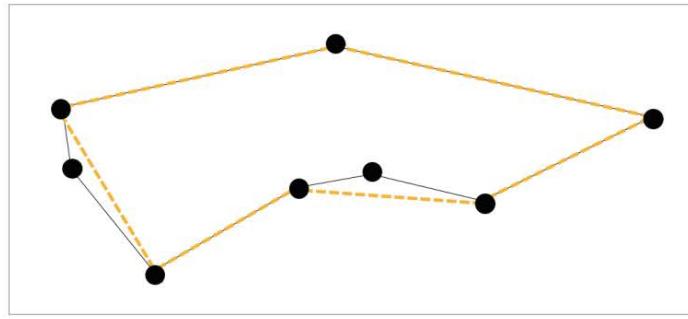
(c)



(d)



(e)



(f)

그림 7.8 더글라스-패커 알고리즘

예제 7.5 C# OpenCvSharp에서의 다각형 근사

```
using System;
using OpenCvSharp;

namespace Project
{
    class Program
    {
        static void Main(string[] args)
        {
            Mat src = Cv2.ImRead("chess.png");
            Mat gray = new Mat();
            Mat binary = new Mat();
            Mat morph = new Mat();
            Mat image = new Mat();
            Mat dst = src.Clone();

            Mat kernel = Cv2.GetStructuringElement(MorphShapes.Rect, new Size(3, 3));
            Point[][] contours;
            HierarchyIndex[] hierarchy;

            Cv2.CvtColor(src, gray, ColorConversionCodes.BGR2GRAY);
            Cv2.Threshold(gray, binary, 230, 255, ThresholdTypes.Binary);
            Cv2.MorphologyEx(binary, morph, MorphTypes.Close, kernel, new Point(-1, -1), 2);
            Cv2.BitwiseNot(morph, image);
```

```
Cv2.FindContours(image, out contours, out hierarchy, RetrievalModes.Tree,
ContourApproximationModes.ApproxTC89KCOS);

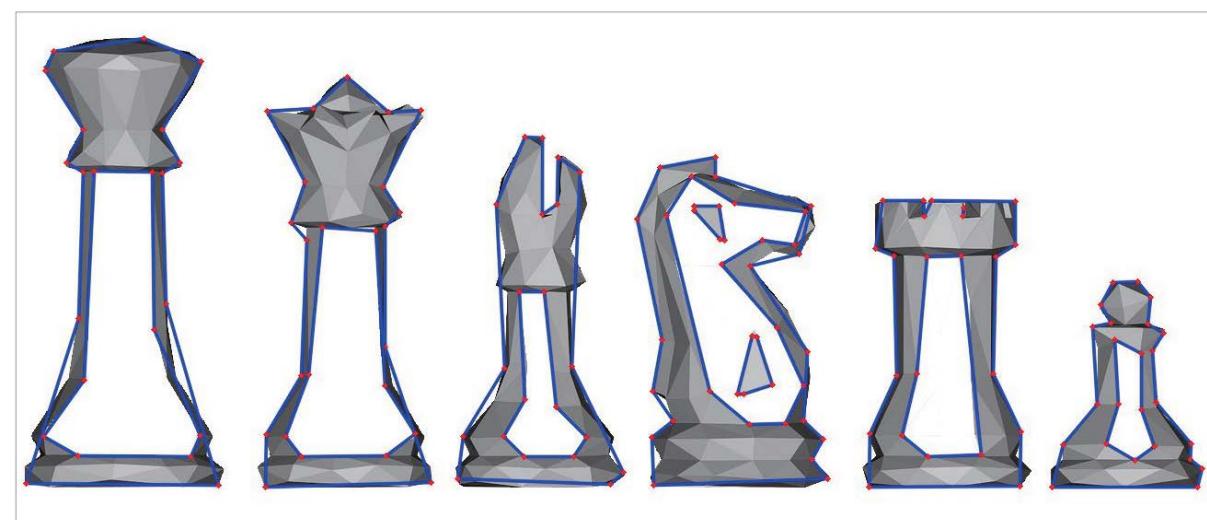
for (int i = 0; i < contours.Length; i++)
{
    double perimeter = Cv2.ArcLength(contours[i], true);
    double epsilon = perimeter * 0.01;

    Point[] approx = Cv2.ApproxPolyDP(contours[i], epsilon, true);
    Point[][] draw_approx = new Point[][] { approx };
    Cv2.DrawContours(dst, draw_approx, -1, new Scalar(255, 0, 0), 2,
LineTypes.AntiAlias);

    for (int j = 0; j < approx.Length; j++)
    {
        Cv2.Circle(dst, approx[j], 1, new Scalar(0, 0, 255), 3);
    }
}

Cv2.ImShow("dst", dst);
Cv2.WaitKey(0);
Cv2.DestroyAllWindows();
}
}
```

【출력 결과】



예제 7.6 Python OpenCV에서의 다각형 근사

```
import cv2

src = cv2.imread("chess.png")
dst = src.copy()

kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (3, 3))

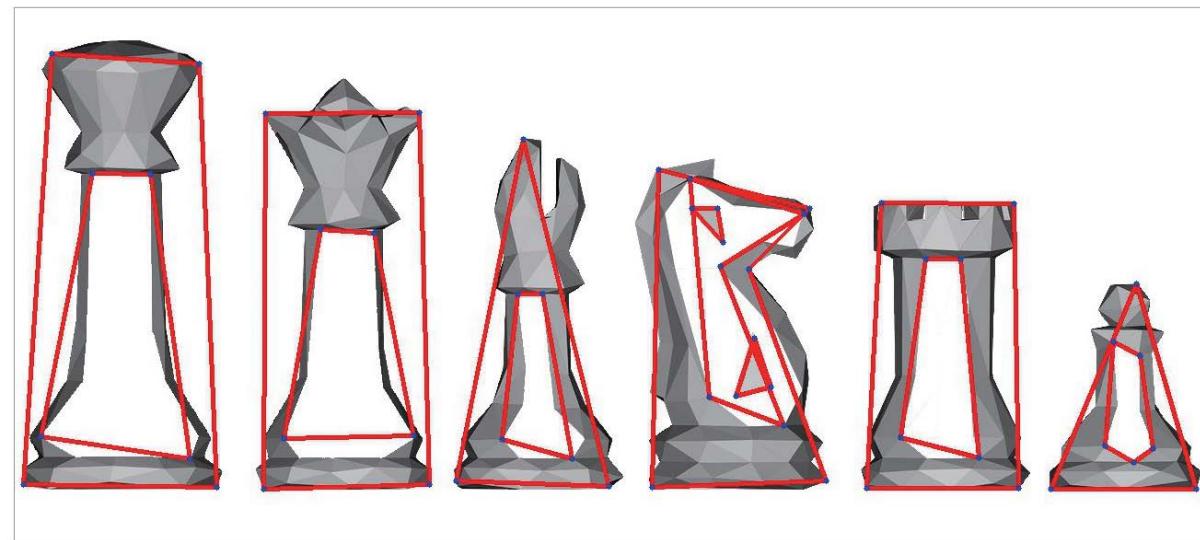
gray = cv2.cvtColor(src, cv2.COLOR_RGB2GRAY)
ret, binary = cv2.threshold(gray, 230, 255, cv2.THRESH_BINARY)
morp = cv2.morphologyEx(binary, cv2.MORPH_CLOSE, kernel, iterations=2)
image = cv2.bitwise_not(morp)

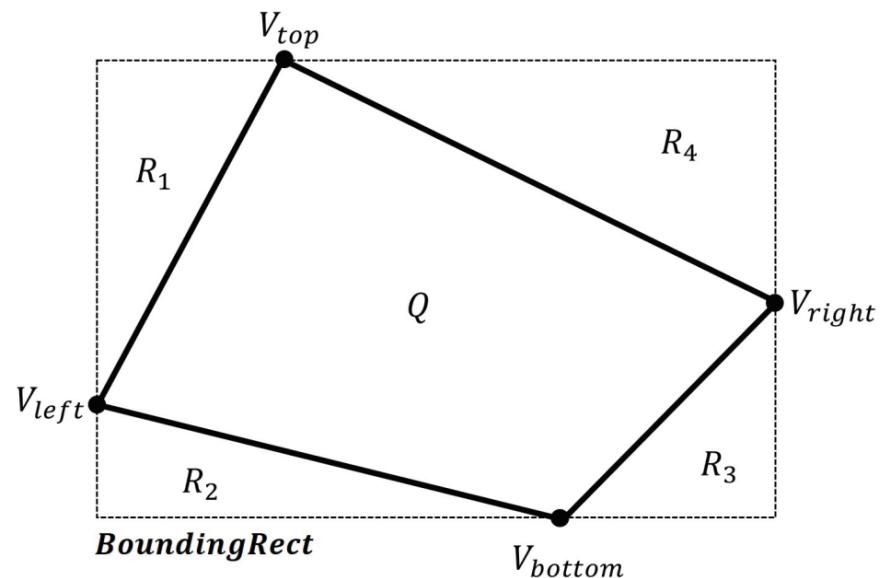
contours, hierarchy = cv2.findContours(image, cv2.RETR_TREE, cv2.CHAIN_APPROX_NONE)

for i in contours:
    perimeter = cv2.arcLength(i, True)
    epsilon = perimeter * 0.05
    approx = cv2.approxPolyDP(i, epsilon, True)
    cv2.drawContours(dst, [approx], 0, (0, 0, 255), 3)
    for j in approx:
        cv2.circle(dst, tuple(j[0]), 3, (255, 0, 0), -1)

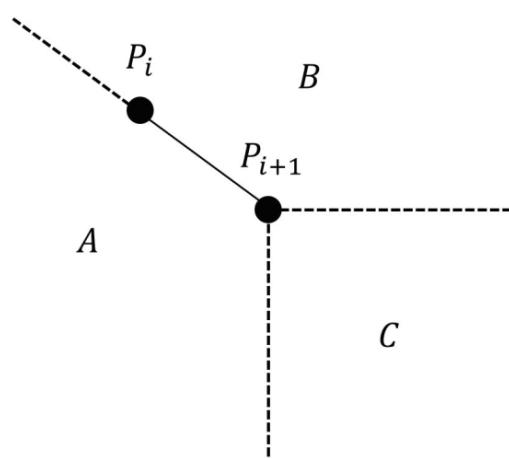
cv2.imshow("dst", dst)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

[출력 결과]





(a) 정점 검출



(b) 볼록점 검출

그림 7.9 스크랜스키 알고리즘

공간 모멘트(spatial moments):

$$m_{ij} = \sum_{x, y} (\text{array}(x, y) \times x^i y^i)$$

중심 모멘트(central moments):

$$mu_{ij} = \sum_{x,y} (array(x, y) \times (x - \bar{x})^i (y - \bar{y})^j)$$

정규화된 중심 모멘트(normalized central moments):

$$m_{ij} = \frac{m_{ij}}{m_{00}^{\frac{(i+j)}{2} + 1}}$$

질량 중심(mass center):

$$\bar{x} = \frac{m_{10}}{m_{00}} \quad \bar{y} = \frac{m_{01}}{m_{00}}$$

예제 7.7 C# OpenCvSharp에서의 코너 검출 및 코너 픽셀 세밀화

```
using System;
using OpenCvSharp;

namespace Project
{
    class Program
    {
        static void Main(string[] args)
        {
            Mat src = Cv2.ImRead("dummy.jpg");
            Mat gray = new Mat();
            Mat dst = src.Clone();

            Cv2.CvtColor(src, gray, ColorConversionCodes.BGR2GRAY);

            Point2f[] corners = Cv2.GoodFeaturesToTrack(gray, 100, 0.03, 5, null, 3, false, 0);
            Point2f[] sub_corners = Cv2.CornerSubPix(gray, corners, new Size(3, 3), new Size(-1,
-1), TermCriteria.Both(10, 0.03));

            for (int i = 0; i < corners.Length; i++)
            {
                Point pt = new Point((int)corners[i].X, (int)corners[i].Y);
                Cv2.Circle(dst, pt, 5, Scalar.Yellow, Cv2.FILLED);
            }
        }
    }
}
```

```
for (int i = 0; i < sub_corners.Length; i++)
{
    Point pt = new Point((int)sub_corners[i].X, (int)sub_corners[i].Y);
    Cv2.Circle(dst, pt, 5, Scalar.Red, Cv2.FILLED);
}

Cv2.ImShow("dst", dst);
Cv2.WaitKey(0);
Cv2.DestroyAllWindows();
}

}
```

【출력 결과】



예제 7.8 Python OpenCV에서의 코너 검출 및 코너 픽셀 세밀화

```
import cv2

src = cv2.imread("dummy.jpg")
dst = src.copy()

gray = cv2.cvtColor(src, cv2.COLOR_RGB2GRAY)
corners = cv2.goodFeaturesToTrack(gray, 100, 0.01, 5, blockSize=3, useHarrisDetector=True, k=0.03)

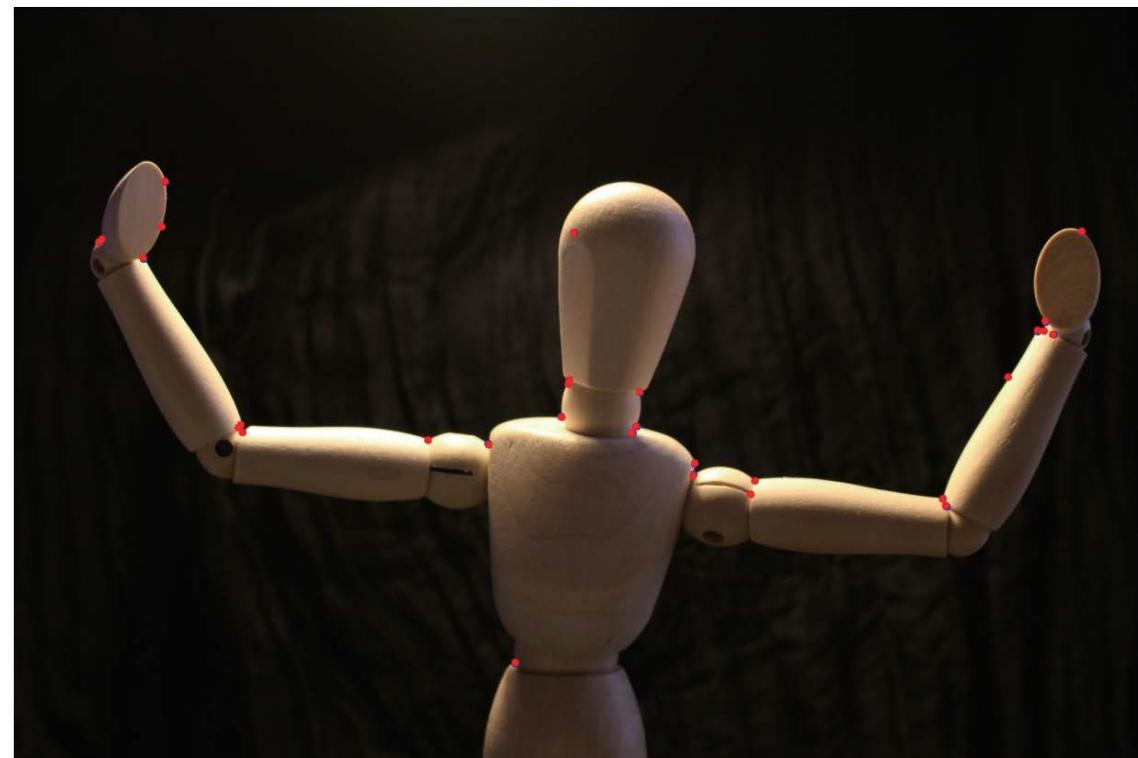
for i in corners:
    cv2.circle(dst, tuple(i[0]), 3, (255, 0, 0), 5)

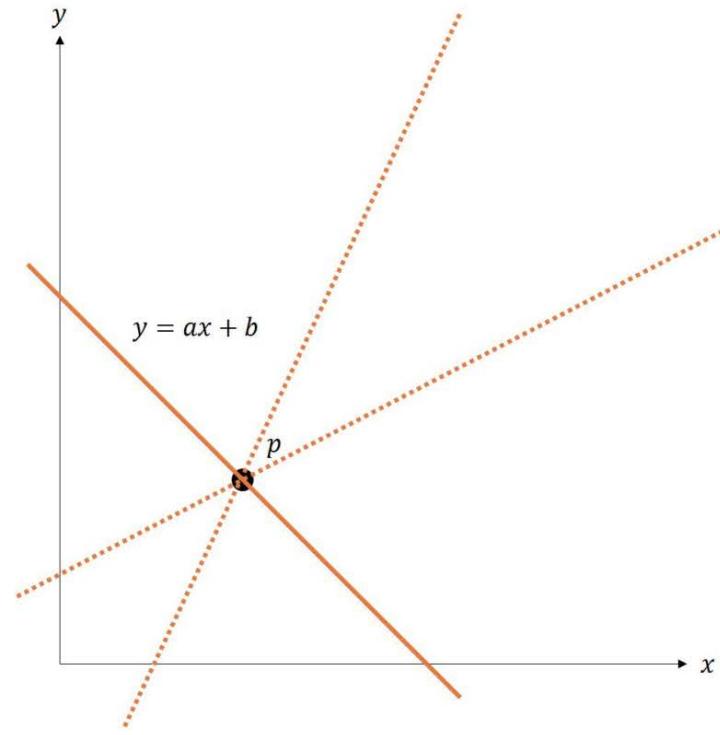
criteria = (cv2.TERM_CRITERIA_MAX_ITER + cv2.TERM_CRITERIA_EPS, 30, 0.001)
cv2.cornerSubPix(gray, corners, (5, 5), (-1, -1), criteria)

for i in corners:
    cv2.circle(dst, tuple(i[0]), 3, (0, 0, 255), 5)

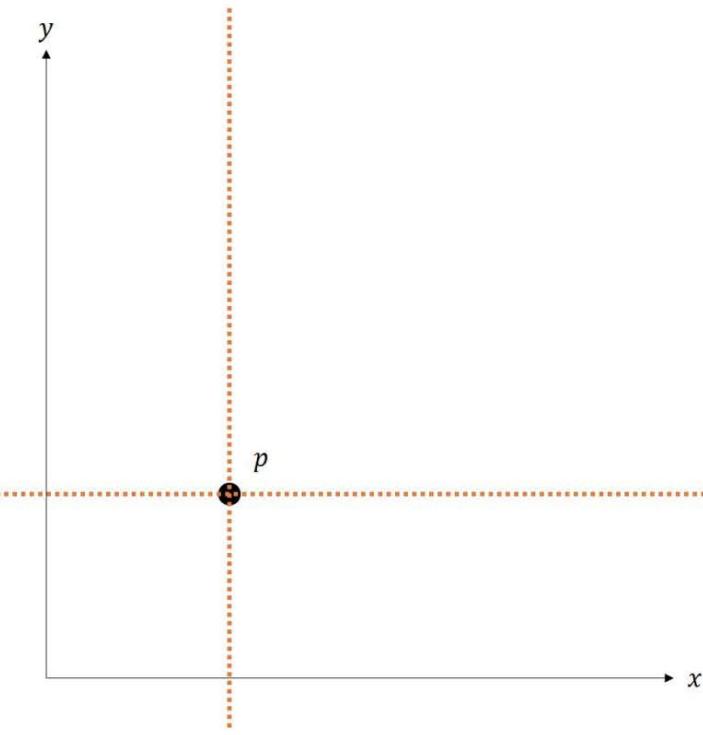
cv2.imshow("dst", dst)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

[출력 결과]





(a) 직선의 방정식

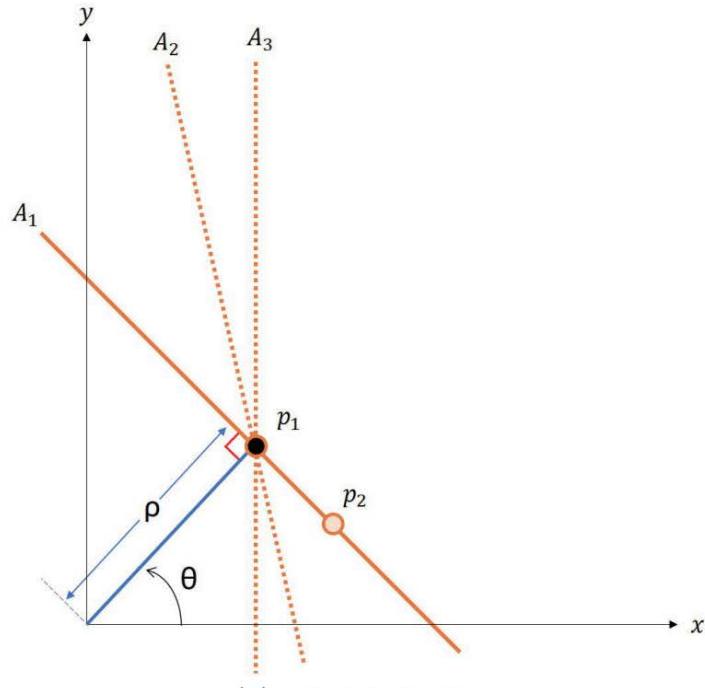


(b) 기울기의 한계

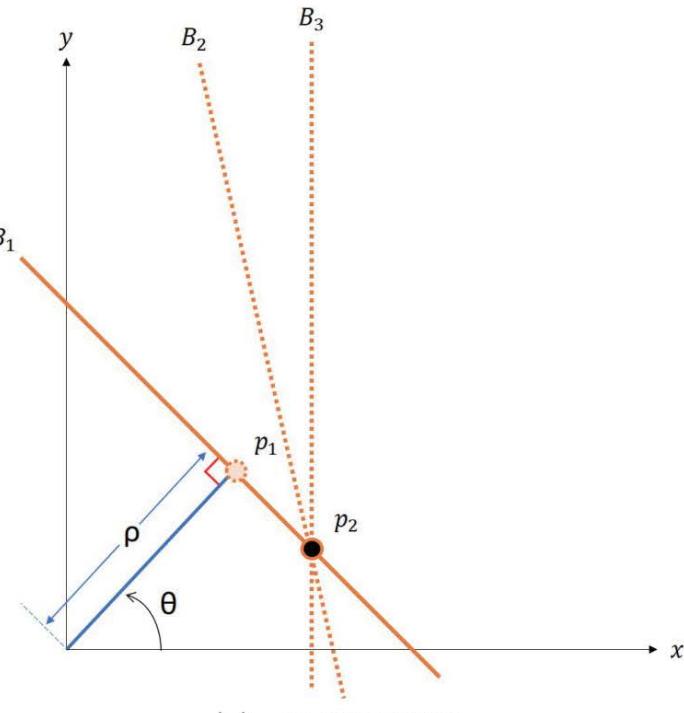
그림 7.10 기울기와 절편의 표현식

삼각 함수를 활용한 직선의 방정식:

$$\rho = x \sin \theta + y \cos \theta$$



(a) p_1 에 대한 방정식



(b) p_2 에 대한 방정식

그림 7.11 거리와 각도의 표현식 예시

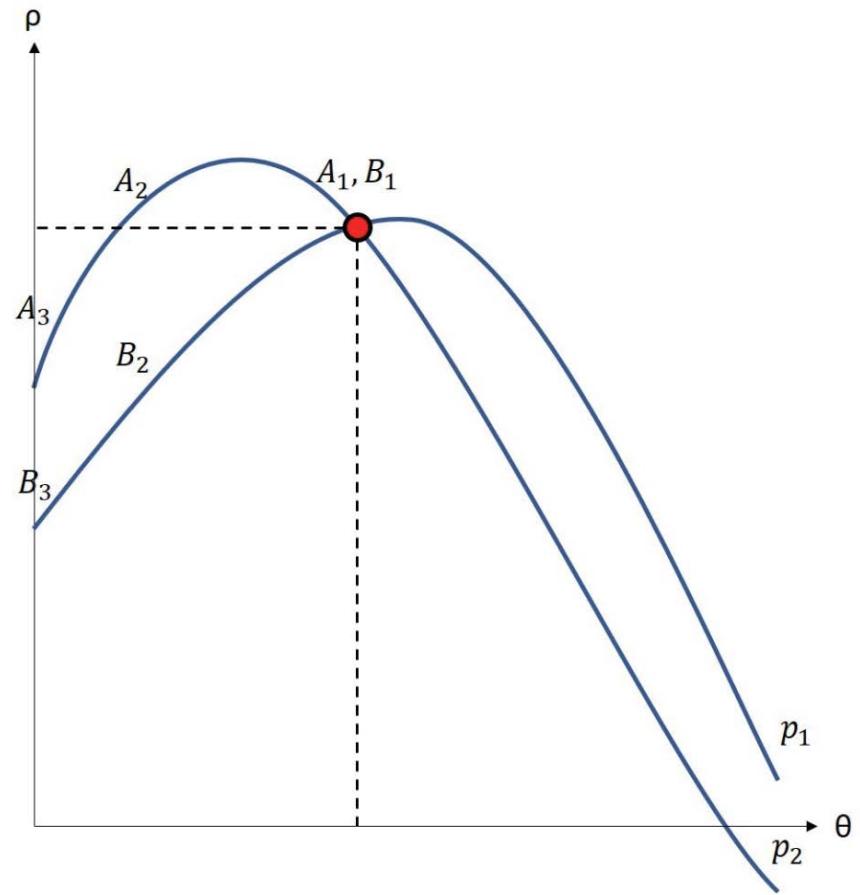


그림 7.12 극좌표 표현식 예시

예제 7.9 C# OpenCvSharp에서의 확률 허프 변환

```
using System;
using OpenCvSharp;

namespace Project
{
    class Program
    {
        static void Main(string[] args)
        {
            Mat src = Cv2.ImRead("card.jpg");
            Mat gray = new Mat();
            Mat binary = new Mat();
            Mat morph = new Mat();
            Mat canny = new Mat();
            Mat dst = src.Clone();

            Mat kernel = Cv2.GetStructuringElement(MorphShapes.Rect, new Size(3, 3));

            Cv2.CvtColor(src, gray, ColorConversionCodes.BGR2GRAY);
            Cv2.Threshold(gray, binary, 150, 255, ThresholdTypes.Binary);
            Cv2.Dilate(binary, morph, kernel, new Point(-1, -1));
            Cv2.Erode(morph, morph, kernel, new Point(-1, -1), 3);
            Cv2.Dilate(morph, morph, kernel, new Point(-1, -1), 2);
            Cv2.Canny(morph, canny, 0, 0, 3);
```

```
LineSegmentPoint[] lines = Cv2.HoughLinesP(canny, 1, Cv2.PI/180, 140, 50, 10);

for (int i=0; i < lines.Length; i++)
{
    Cv2.Line(dst, lines[i].P1, lines[i].P2, Scalar.Yellow, 2);
}

Cv2.ImShow("dst", dst);
Cv2.WaitKey(0);
Cv2.DestroyAllWindows();
}

}
```

[출력 결과]



예제 7.10 Python OpenCV에서의 허프 변환

```
import numpy as np
import cv2

src = cv2.imread("card.jpg")
dst = src.copy()

kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (3, 3), (-1, -1))

gray = cv2.cvtColor(src, cv2.COLOR_BGR2GRAY)
_, binary = cv2.threshold(gray, 150, 255, cv2.THRESH_BINARY)
morp = cv2.dilate(binary, kernel)
morp = cv2.erode(morp, kernel, iterations=3)
morp = cv2.dilate(morp, kernel, iterations=2)
canny = cv2.Canny(morp, 0, 0, apertureSize=3, L2gradient=True)

lines = cv2.HoughLines(canny, 1, np.pi/180, 140, srn=50, stn=10, min_theta=0, max_theta=np.pi/2)

for i in lines:
    rho, theta = i[0][0], i[0][1]
    a, b = np.cos(theta), np.sin(theta)
    x0, y0 = a*rho, b*rho

    scale = src.shape[0] + src.shape[1]

    x1 = int(x0 + scale * -b)
```

```
y1 = int(y0 + scale * a)
x2 = int(x0 - scale * -b)
y2 = int(y0 - scale * a)

cv2.line(dst, (x1, y1), (x2, y2), (0, 255, 255), 2)
cv2.circle(dst, (x0, y0), 3, (255, 0, 0), 5, cv2.FILLED)

cv2.imshow("dst", dst)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

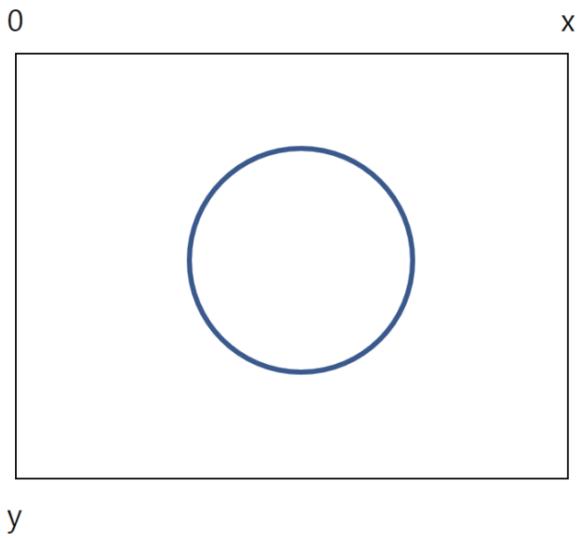
【출력 결과】



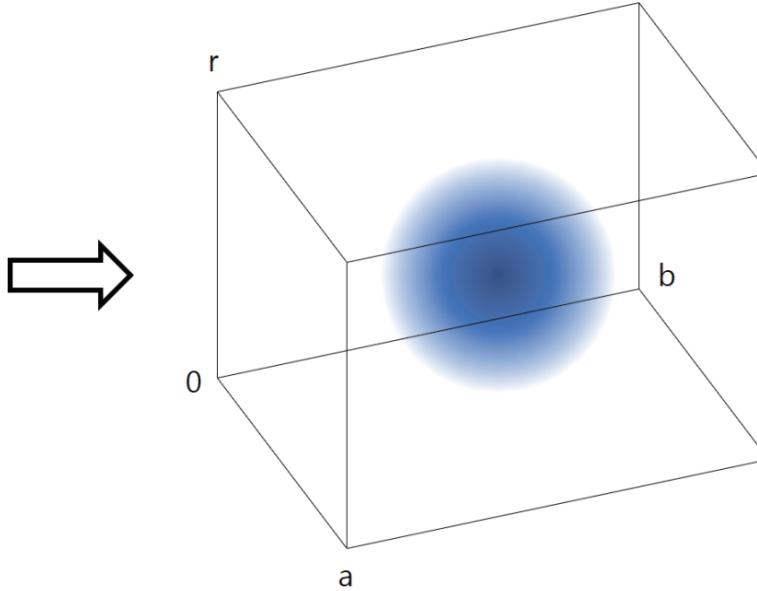
원의 방정식:

$$(x - a)^2 + (y - b)^2 = r^2$$

$$a = x_{center}, b = y_{center}$$



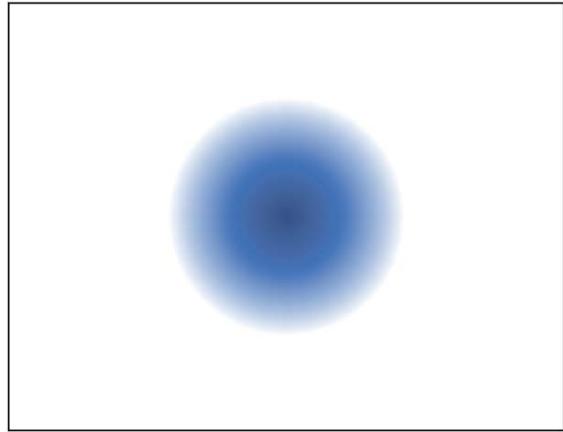
Edge Detection



3D Histogram

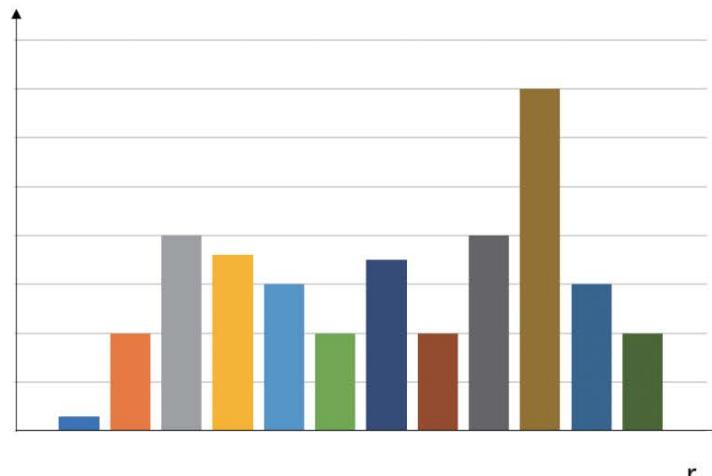
그림 7.13 표준 허프 원 변환

b



0 a

2D Histogram



1D Histogram

그림 7.14 2단계 헤프 원 변환

예제 7.11 C# OpenCvSharp에서의 허프 원 변환

```
using System;
using OpenCvSharp;

namespace Project
{
    class Program
    {
        static void Main(string[] args)
        {
            Mat src = Cv2.ImRead("colorball.png");
            Mat image = new Mat();
            Mat dst = src.Clone();

            Mat kernel = Cv2.GetStructuringElement(MorphShapes.Rect, new Size(3, 3));

            Cv2.CvtColor(src, image, ColorConversionCodes.BGR2GRAY);
            Cv2.Dilate(image, image, kernel, new Point(-1, -1), 3);
            Cv2.GaussianBlur(image, image, new Size(13, 13), 3, 3, BorderTypesReflect101);
            Cv2.Erode(image, image, kernel, new Point(-1, -1), 3);

            CircleSegment[] circles = Cv2.HoughCircles(image, HoughMethods.Gradient, 1, 100, 100,
35, 0, 0);

            for (int i = 0; i < circles.Length; i++)
            {
```

```
    Point center = new Point(circles[i].Center.X, circles[i].Center.Y);

    Cv2.Circle(dst, center, (int)circles[i].Radius, Scalar.White, 3);
    Cv2.Circle(dst, center, 5, Scalar.AntiqueWhite, Cv2.FILLED);
}

Cv2.ImShow("dst", dst);
Cv2.WaitKey(0);
Cv2.DestroyAllWindows();
}
}
```

[출력 결과]



예제 7.12 Python OpenCV에서의 허프 원 변환

```
import cv2

src = cv2.imread("colorball.png")
dst = src.copy()

image = cv2.cvtColor(src, cv2.COLOR_BGR2GRAY)

circles = cv2.HoughCircles(image, cv2.HOUGH_GRADIENT, 1, 100, param1=100, param2=35, minRadius=80,
maxRadius=120)

for i in circles[0]:
    cv2.circle(dst, (i[0], i[1]), i[2], (255, 255, 255), 5)

cv2.imshow("dst", dst)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

【출력 결과】

