

# <RO:BIT ROS2 - 4일차 과제1, 2 보고서>

---

작성자: 박기홍

직책: 19기 예비 단원

소속: RO:BIT 지능 팀

작성일: 2024년 11월 4일

## 목차

1. 요약
2. 관련 이론 및 개념
3. 참고 문헌

## 1. 요약

이 보고서는 OpenCV에 대한 보고서로, OpenCV 관련 함수와 기능, 원리에 대해 공부한 내용을 포함하고 있다.

## 2. 관련 이론 및 개념

### 2.1. ROS Parameter

ROS 파라미터는 노드 내부의 파라미터 서버를 통해 외부의 파라미터 클라이언트와 통신하여 다양한 파라미터를 설정할 수 있다. 각 노드는 고유한 파라미터 서버를 가지며, 이를 통해 노드의 내부 및 외부에서 손쉽게 파라미터를 지정하거나 변경할 수 있다.

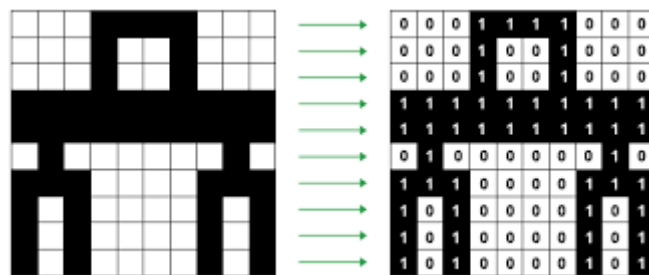
#### 2.1.1. 명령어 예시

```
$ ros2 param list: 각 노드의 파라미터를 확인
$ ros2 param get <노드명> <파라미터명>: 특정 파라미터 값을 불러옴
$ ros2 param set <노드명> <파라미터명> <값>: 파라미터 값을 설정
$ ros2 param dump <노드명> --output-dir <경로>: 파라미터 파일 저장
```

### 2.2. OpenCV 색상 형식

OpenCV는 다양한 이미지 색상 형식을 제공한다. 각 색상 형식에 따라 이미지의 표현 방식이 다르다.

- **Grayscale:** 각 픽셀이 0~255 사이의 값을 가지며, 흰색(255)과 검정색(0)을 표현
- **Binary:** 0과 255로만 구성된 흑백 이미지



[Binary Image]

- **RGB:** 각 픽셀이 RED, GREEN, BLUE 세 가지 값을 통해 컬러를 표현
- **HSV:** 색상(Hue), 채도(Saturation), 명도(Value)로 색을 표현하여 색 판단이 용이

## 2.3.OpenCV 이미지 처리 기능

이미지 전처리 및 후처리를 위해 다양한 OpenCV 기능을 사용할 수 있다.

- **가우시안 블러**: 이미지의 노이즈를 줄이기 위해 가우시안 함수 적용
- **침식(Erode) 및 팽창(Dilate)**: 이미지의 특정 요소를 강조하거나 줄이기 위해 사용
- **관심영역(ROI)**: 이미지의 특정 영역을 마스크 처리하여 강조
- **Canny**: 이미지 엣지 감지



[Canny가 적용된 이미지]

- **Houghline**: 이미지에 직선을 표시하여 선 형태 감지

## 2.4 OpenCV 이미지 처리 원리

### 2.4.1. 가우시안 블러(cv::GaussianBlur)

가우시안 블러는 이미지의 노이즈를 제거하거나 부드럽게 만들기 위해 사용되는 필터링 기법이다.

가우시안 함수라는 수학적 모델을 이용하여 이미지의 중심에서 멀어질수록 픽셀 값의 영향이 줄어들게 만든다고 한다.

#### 2.4.1.1. 원리

가우시안 분포 함수에 기반하여 주변 픽셀의 가중 평균을 계산한다. 이미지의 중앙 픽셀은 영향을 많이 받고, 주변으로 갈수록 적게 받아 부드러운 블러링이 적용되는 원리다.

#### 2.4.1.2. 구조

```
cv::GaussianBlur(src, dst, ksize, sigmaX, sigmaY);
```

- **src**: 입력된 이미지다.
- **dst**: 가우시안 블러가 적용된 결과 이미지다.
- **ksize**: 필터 커널 크기(예: cv::Size(3,3)). 홀수여야 하고, 크기가 클수록 블러 효과가 강해진다.
- **sigmaX, sigmaY**: X, Y 방향의 표준 편차. sigmaY를 0으로 설정하면 sigmaX와 동일하다.

#### 2.4.1.3. 예제

```
cv::Mat input_image; // 입력 이미지
cv::Mat blurred_image;
cv::GaussianBlur(input_image, blurred_image, cv::Size(7, 7), 1.5);
```

### 2.4.2. 침식과 팽창 (cv::erode, cv::dilate)

erode와 dilate는 이미지의 경계를 줄이거나 확장하는 데 사용된다.

#### 2.4.2.1. 원리

침식(Erosion): 커널이 적용될 때, 객체 영역이 줄어들도록 흰색 픽셀을 검정색으로 변경한다.

팽창(Dilation): 커널이 적용될 때, 객체 영역이 확장되도록 검정색 픽셀을 흰색으로 변경한다.

#### 2.4.2.2. 구조

**cv::erode(침식)**

```
cv::erode(src, dst, kernel, anchor, iterations, borderType,
borderValue);
```

- **src**: 입력 이미지.

- **dst**: 침식이 적용된 결과 이미지.
- **kernel**: 구조 요소로 사용할 커널. 크기와 형태에 따라 침식 효과가 달라지며, `cv::getStructuringElement`로 생성할 수 있다.
- **anchor**: 커널의 기준점. 기본값은 `(-1, -1)`로 중앙을 의미한다.
- **iterations**: 침식 반복 횟수.
- **borderType**: 가장자리 픽셀 처리 방식.
- **borderValue**: 가장자리 픽셀 값.

### **cv::dilate(팽창)**

```
cv::dilate(src, dst, kernel, anchor, iterations, borderType,
borderValue);
```

#### **2.4.2.3. 예제**

```
cv::Mat kernel = cv::getStructuringElement(cv::MORPH_RECT, cv::Size(5,
5));
cv::erode(input_image, eroded_image, kernel);
cv::dilate(eroded_image, dilated_image, kernel);
```

### **2.4.3. 관심영역 (cv::Rect roi)**

관심영역은 이미지에서 특정 영역을 추출하기 위해 사용되는 사각형 객체이다. 관심영역(ROI)을 설정하여 해당 부분만 집중적으로 처리하거나 분석할 수 있다.

#### **2.4.3.1. 원리**

이미지의 특정 부분만 마스크 처리하여 관심 영역으로 설정한다. 관심영역 외부의 데이터는 무시되는 원리이다.

#### **2.4.3.2. 구조**

```
cv::Rect roi(x, y, width, height);
```

- **cv::Rect**: OpenCV에서 사각형을 지정하는 클래스
- **roi**: 객체 이름
- **x, y**: ROI의 왼쪽 상단 모서리 좌표.
- **width, height**: ROI의 너비와 높이.

#### 2.4.3.3. 예제

```
cv::Rect roi(50, 50, 200, 200); // x, y 좌표와 width, height 설정
cv::Mat roi_image = input_image(roi); // input_image에서 지정된 ROI
영역을 잘라내어 roi_image에 저장
```

#### 2.4.4. Canny 엣지 검출 (cv::Canny)

Canny는 이미지의 경계선을 감지하기 위한 엣지 검출 메서드이다. 이미지에서 중요한 윤곽선만을 추출하여 객체를 더욱 뚜렷하게 구분할 수 있다.

##### 2.4.4.1. 원리

1. 노이즈 제거(가우시안 블러 사용)
2. 이미지 기울기 계산
3. 비최대 억제: 가장 강한 엣지만 유지
4. 이중 임계값을 통해 강한 엣지와 약한 엣지를 구분하여 엣지를 감지

##### 2.4.4.2. 구조

```
cv::Canny(src, dst, threshold1, threshold2, apertureSize, L2gradient);
```

- **src**: 입력 이미지.
- **dst**: 엣지 검출 결과 이미지.
- **threshold1, threshold2**: 낮은 임계값과 높은 임계값으로, 엣지로 판단할 픽셀 경계를 설정한다.
- **apertureSize**: 소벨 연산에서 커널의 크기 (3, 5, 7 등 홀수여야 함). 기본값은 3이다.



- **L2gradient**: true로 설정하면 정확한 L2 norm을 사용해 경계를 계산한다. 기본값은 false이다.

#### 2.4.4.3. 예제

```
cv::Canny(input_image, edges, 100, 200); // 임계값 100과 200을 설정
```

### 2.4.5. HoughLinesP(cv::HoughLinesP)

HoughLinesP는 이미지에서 직선을 검출하는 함수로, 특정 길이와 각도를 가진 직선만 검출할 수 있다.

#### 2.4.5.1. 원리

이미지의 직선 후보를 매개변수 공간에서 투표 방식으로 검출한다. 확률적 허프 변환을 사용하여 계산 속도를 높이고, 지정한 최소 길이와 최대 간격을 기준으로 선분을 찾는 원리이다.

#### 2.4.5.2. 구조

```
cv::HoughLinesP(src, lines, rho, theta, threshold, minLineLength, maxLineGap);
```

- **src**: 엣지 검출이 된 이미지.
- **lines**: 검출된 선을 저장할 벡터.
- **rho**: 매개변수의 거리 해상도(픽셀 단위).
- **theta**: 각도 해상도(라디안 단위).
- **threshold**: 선으로 간주될 최소 교차점 수.
- **minLineLength**: 검출할 직선의 최소 길이.
- **maxLineGap**: 직선으로 인식할 점 사이의 최대 거리.

#### 2.4.5.3. 예제

```
cv::Mat edges; // 엣지 검출 결과를 저장할 행렬 생성
```

```
cv::Canny(input_image, edges, 100, 200); // Canny 엣지 검출 수행 (임계값
100과 200 설정)
std::vector<cv::Vec4i> lines; // 검출된 선분을 저장할 벡터 (각 선분은
시작점과 끝점의 좌표를 가진 Vec4i 형태)
cv::HoughLinesP(edges, lines, 1, CV_PI / 180, 50, 50, 10); // 확률적
허프 변환을 사용하여 선분을 검출
```

**\*cv::Vec4i**는 OpenCV에서 사용하는 벡터 형식 중 하나이다. 4개의 정수를 저장할 수 있는 자료형이다. 선분의 시작점과 끝점 좌표를 저장하는 데 자주 사용된다. 시작점(x1, y1), 끝점(x2, y2)

### 3. 참고 문헌

- 버건디 팩토리. “[OpenCV] cv::Gaussian Blur 함수 사용하기”.Tistory.  
<https://nowtimeisthat.tistory.com/16>
- OpenCV. “Image Filtering”,  
[https://docs.opencv.org/4.5.2/d4/d86/group\\_imgproc\\_filter.html#gaabe8c836e97159a9193fb0b11ac52cf1](https://docs.opencv.org/4.5.2/d4/d86/group_imgproc_filter.html#gaabe8c836e97159a9193fb0b11ac52cf1)
- 김진혁, “6. OpenCV C++ 이미지 침식, 팽창 (erode, dilate) 를 통해 노이즈 제거, 없는 부분 채우기”, <https://m.blog.naver.com/kristoper12/221860061025>
-