

바닥 패턴의 특징점 추출

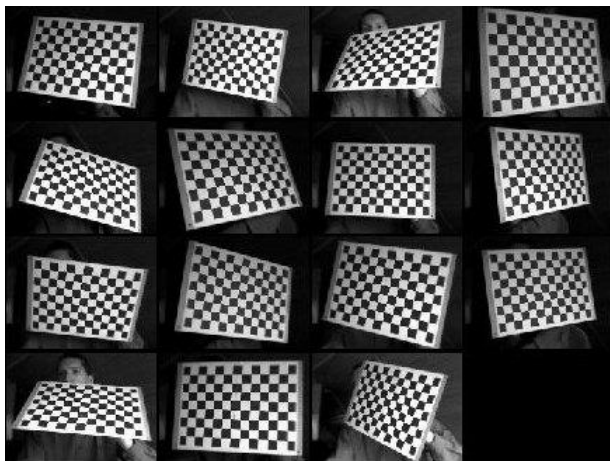
바닥 패턴의 특징점 추출

KITECH 양광웅 작성

Camera Calibration

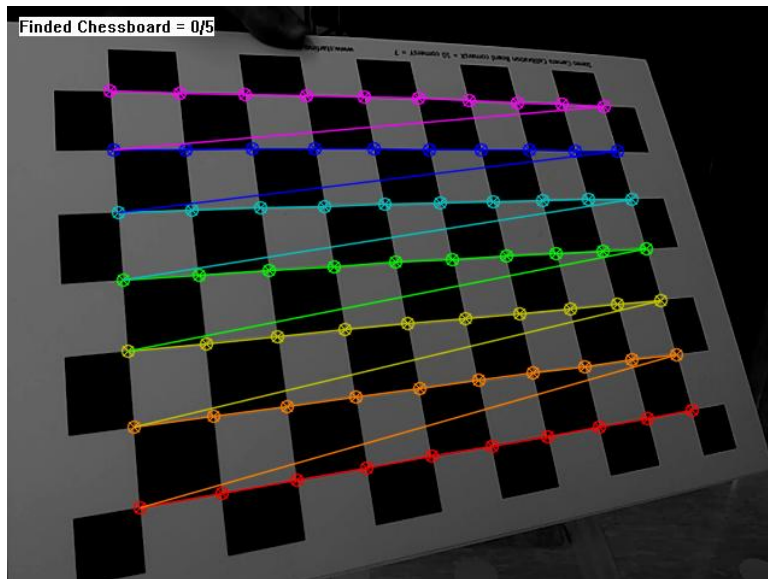
카메라 렌즈의 제조과정과 CMOS 칩의 조립 과정에서 발생하는 오차로 인해 카메라의 입력 영상은 왜곡되어 CMOS 센서에 상이 맺히게 된다. 방사 왜곡(radial distortion)은 렌즈의 모양에 의해 발생하고, 접선 왜곡(tangential distortion)은 카메라 조립 과정에서 발생하게 된다.

이 두 왜곡을 보정하기 위해 OpenCV의 캘리브레이션 함수 `cvCalibrateCamera2()`를 사용한다. 먼저 다음 그림에서와 같이 여러 장의 체스 판을 찍은 사진이 필요하다.



체스 판의 사진이 주어지면 `cvFindChessboardCorners()` 함수로 체스판 코너를 찾는다. 찾은 코너는 pixel 단위로 구해지기 때문에 `cvFindCornerSubPix()` 함수로 코너의 위치를 서브픽셀 정확도로 향상한다.

다음 그림은 찾은 코너들을 `cvDrawChessboardCorners()` 함수로 그린 영상이다.



이제 이미지 상의 코너의 위치와 실제 체스판의 코너 위치간의 관계를 이용하여 카메라의 내부 행렬(*intrinsic matrix*)과 왜곡 계수(*distortion coefficient*)를 계산한다. 이 과정은 복잡하지만 `cvCalibrateCamera2()` 함수를 이용하면 쉽게 할 수 있다.

캘리브레이션 함수가 계산한 내부 행렬과 왜곡 계수를 다른 프로그램에서도 사용할 수 있도록 `cvSave()` 함수로 `Intrinsics.xml` 파일과 `Distortion.xml` 파일로 저장한다.

Camera Undistortion

카메라 캘리브레이션 과정으로 내부 행렬과 왜곡 계수를 계산하였다면, 이제 이 두 값을 이용하여 이미지의 왜곡을 제거해야 한다.

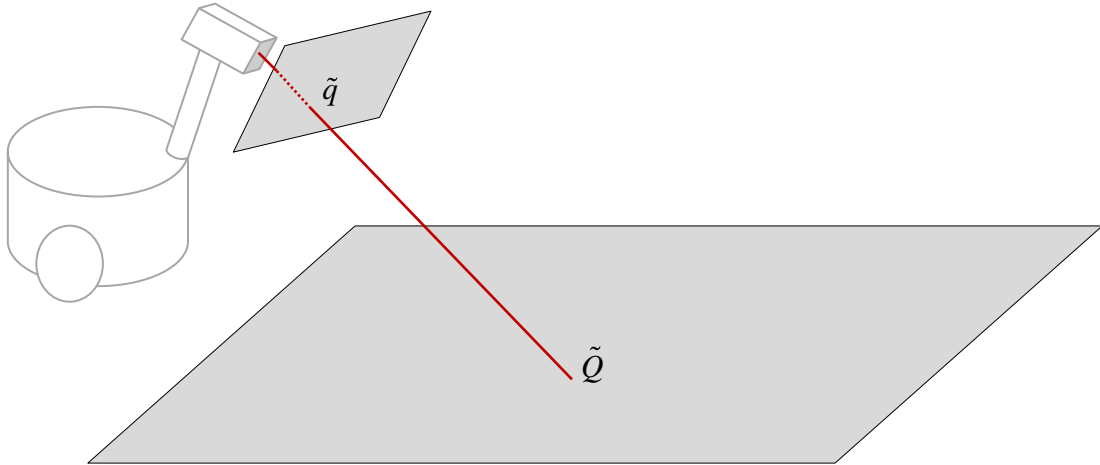
먼저 캘리브레이션 과정에서 저장했던 `Intrinsics.xml` 파일과 `Distortion.xml` 파일을 `cvLoad()` 함수로 읽어 카메라 내부 행렬과 왜곡 계수를 읽어온다.

그리고 내부 행렬과 왜곡 계수로부터 `cvInitUndistortMap()` 함수를 사용하여 왜곡 제거를 위한 2개의 지도(`MapX`, `MapY`)를 구성한다.

이제 카메라로부터 이미지가 입력될 때마다 `MapX`, `MapY`를 사용하여 왜곡을 제거한 영상을 만든다. 이 과정은 `cvRemap()` 함수로 수행한다.

Planar Homography

컴퓨터 비전에서 평면 호모그래피(planar homography)는 하나의 평면을 다른 평면으로 옮기는 투영 변환을 의미한다. 로봇이 2차원 평면상에서 주행할 때, 2차원 평면(바닥) 상의 점들을 카메라의 영상으로 변환하는 과정도 평면 호모그래피의 한 예이다.



바닥의 한 점 Q 와 이미지 센서 위에 매핑된 점 q 를 각각 동차 좌표계로 표현한다.

$$\tilde{Q} = \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}, \quad \tilde{q} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

그러면 다음과 같은 호모그래피로 표현할 수 있다.

$$\tilde{q} = s\mathbf{H}\tilde{Q}$$

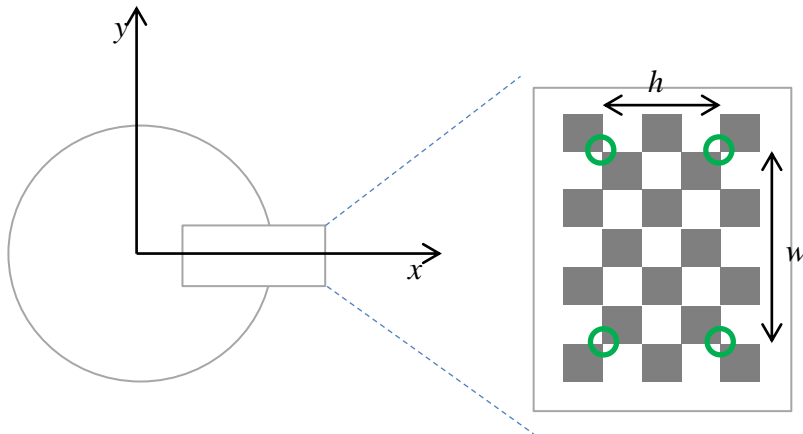
여기서 s 는 스케일 인자이고 \mathbf{H} 는 3x3 호모그래피 행렬이다.

Homography Matrix 계산

호모그래피 행렬을 계산하기 위해 카메라 캘리브레이션 과정에서 사용했던 체스판을 또 사용할 것이다.

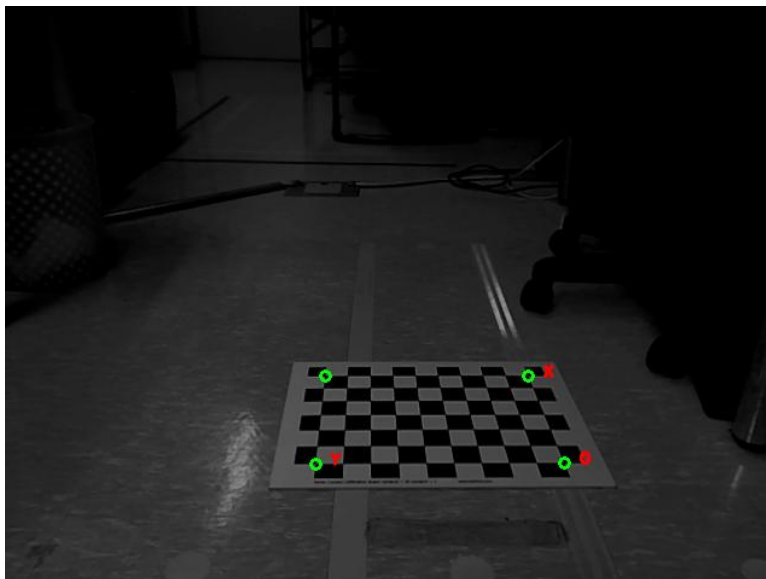
먼저 로봇 좌표계를 기준으로 바닥의 특정 위치에 체스판을 배치한다. 그러면 체스판의 네 모서리의 위치(아래 그림에서 초록색 원으로 표시한 부분)는 로봇 좌표계를 기준으로 쉽게 측정할 수 있다. 예를 들자면, 바닥에 원점을 그리고 로봇의 원점과 일치하도록 로봇을 올려놓은 후 체스판 중심이 바닥 원점으로부터 1m 떨어진 곳에 위치시킨다면, 체스판의 네 모서리 위치는

$(1+h/2, w/2)$, $(1-h/2, w/2)$, $(1+h/2, -w/2)$, $(1-h/2, -w/2)$ 가 됨을 쉽게 알 수 있다.



체스판의 네 모서리를 찾는 기능은 캘리브레이션 과정에서 사용했던 `cvFindChessboardCorners()` 함수와 `cvFindCornerSubPix()` 함수로 모든 코너의 위치를 찾은 후 네 모서리의 코너 위치만 사용하면 된다.

다음 그림은 네 모서리의 위치를 찾은 후 초록색 원으로 표시하고 각각 원점과 X축 방향 위치, Y축 방향 위치를 붉은 색의 O, X, Y 문자로 표시한 것이다.



바닥에 놓인 체스판의 네 모서리 위치와 이미지 상의 체스판의 네 모서리 위치간 관계로부터 변환행렬을 계산한다. 변환 행렬은 `cvGetPerspectiveTransform()` 함수로 구할 수 있다.

변환 행렬은 `cvSave()` 함수로 `Homography.xml` 파일에 저장해 둔다.

Feature Point 추출

여기서는 카메라 입력 영상으로부터 지도의 패턴과 비교하기 위해 특징이 되는 점들을 추출하는 과정을 설명한다.

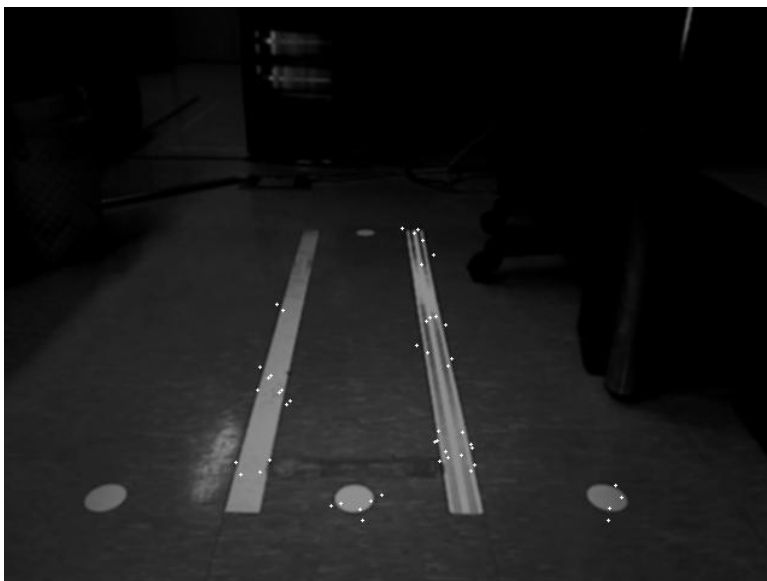
`cvGrabFrame()` 함수와 `cvRetrieveFrame()` 함수로 카메라 영상을 읽어온다.

카메라 입력 영상에서 `cvRemap()` 함수로 왜곡을 제거한 영상을 얻는다.

노이즈에 덜 민감하도록 하기위해 `cvSmooth()` 함수로 가우시안 블러를 적용한다.



상기 영상에서 일정 거리에 있는 두 점을 이미지에 랜덤하게 위치시킨다. 그리고 이 두 점의 값의 차가 주어진 `threshold` 값을 넘어가게 되면 이 두 점을 영상에서 특징점으로 선정한다. 이 과정을 반복하여 여러 개의 특징점을 추출한다.



특징점은 이미지에서 위치와 밝기 정보($x, y, \text{intensity}$)를 가진다.

호모그래피 행렬 \mathbf{H} 을 이용하여 이미지 평면의 특징점의 위치 $\tilde{q} = (u, v, 1)$ 을 로봇 좌표계를 기준으로 하는 바닥면의 한 점 $\tilde{Q} = (X, Y, 1)$ 로 변환한다.

$$\tilde{Q} = \frac{1}{s} \mathbf{H}^{-1} \tilde{q}$$

상기 과정에서 계산한 점 \tilde{Q} 는 로봇 좌표계로 표시된다. 전역 좌표계로 변환하기 위해 로봇의 위치와 방위(x, y, θ)를 고려하면 다음과 같다.

$$\begin{aligned} {}^wQ &= [\mathbf{R}_z \mid \mathbf{t}] \tilde{Q} \\ &= \begin{bmatrix} \cos \theta & -\sin \theta & x \\ \sin \theta & \cos \theta & y \\ 0 & 0 & 1 \end{bmatrix} \tilde{Q} \end{aligned}$$

다음 그림은 카메라 영상에서 인식한 특징점의 위치를 초록색 점으로 지도상에 표시한 것이다.

