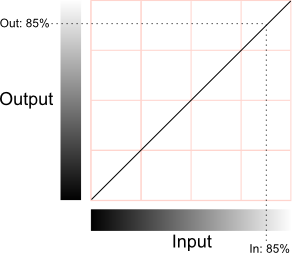
**Red Distortion Effect**

**This Image effect tries to simulate the Red light weakness**: You are not able to differentiate red shades, what is trouble when you must pay attention to brake lights.

In order to obtain this visual effect, we use the Color Correction Curves[[1]](#footnote-1) feature from the Unity Image effects.

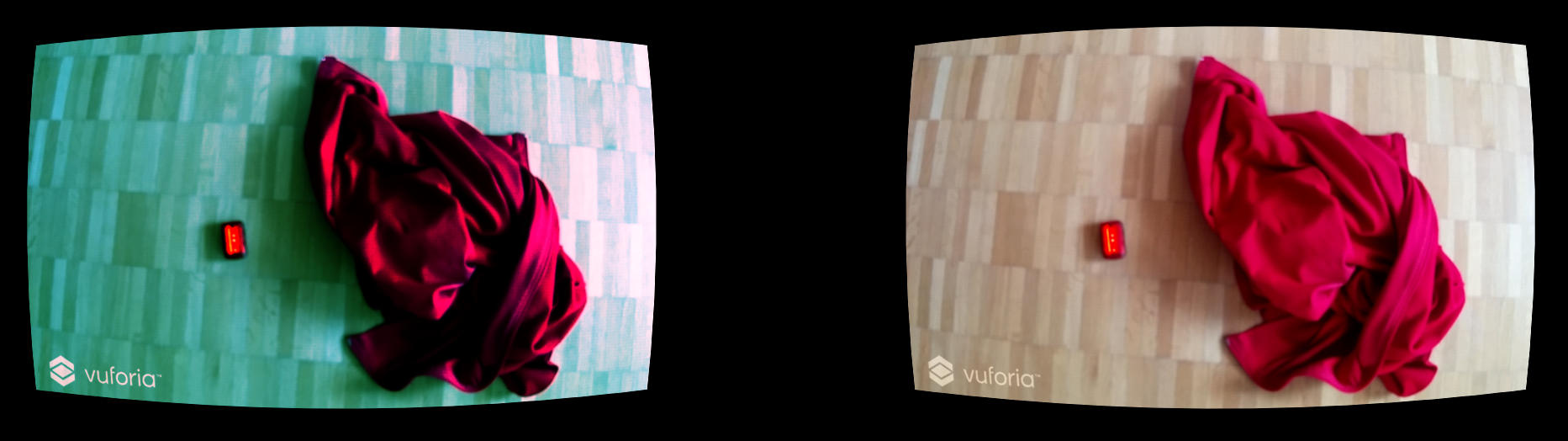
**Color Correction Curves** make color adjustments using curves for each color channel. Depth based adjustments allow you to vary the color adjustment according to a pixel’s distance from the camera.

Curves work on each of the red, green and blue color channels separately and are based around the idea of mapping each input brightness level (i.e., the original brightness value of a pixel) to an output level of your choosing.



The horizontal axis represents the input level and the vertical represents the output level. Any point on the line specifies the output value that a given input is mapped to during processing.

In our case we choose the red channel curve and configured its output in a way that the visibility of the red color is reduced or looks like black. As you can see in the image, it’s difficult to recognize red traffic lights (the light in the image is a bike brake light).

The collateral effect of this curve is the reinforcement of the green color, but ASN told us that this “green effect” complements well the red light weakness because this weakness is stronger during the night. And the green effect gets dark the whole vision, so this small disadvantage becomes an advantage.

1. https://docs.unity3d.com/Manual/script-ColorCorrectionCurves.html [↑](#footnote-ref-1)