Canny Edge Detector je algoritmus na detekciu hrán v obraze, ktorý pozostáva z viacerých algoritmov na detekciu širokého rozsahu hrán v obrazoch. Bol vyvinutý Johnom F. Cannym v roku 1986.

Algoritmus detekcie Canny Edge sa skladá z 5 krokov:

1. Noise reduction – redukcia šumu
2. Gradient calculation – výpočet gradientu
3. Non-maximu suppression
4. Double threshold
5. Edge Tracking by Hysteresis

Po vykonaní týchto krokov získame obraz zo zvýraznenímy hranamy pôvodného obrazu.

## 2.1 Noise reduction - Redukcia šumu

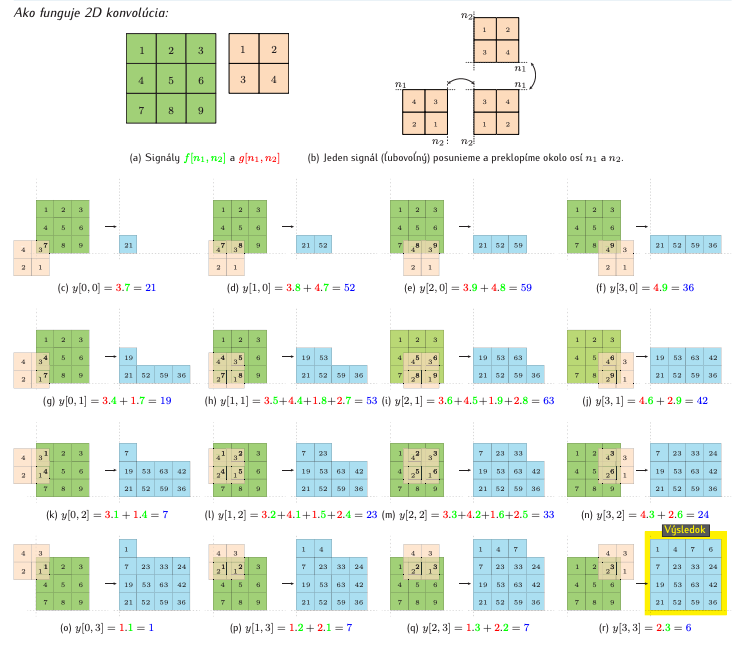
Pri redukcii šumu budeme aplikovať konvolúciu pri, ktorej prechádzame obrazom a na výpočet novej hodnoty jasu pixelu použijeme jeho rôzne veľké okolie. Pri tomto postupe využijeme Gaussov filter a Gaussove jadro (3x3, 5x5, 7x7 atď...). Ako prvé budeme definovať konvolúciu a následne vysvetlíme a zadefinujeme Gassov filter a jadro.

### 2.1.1 Konvolúcia

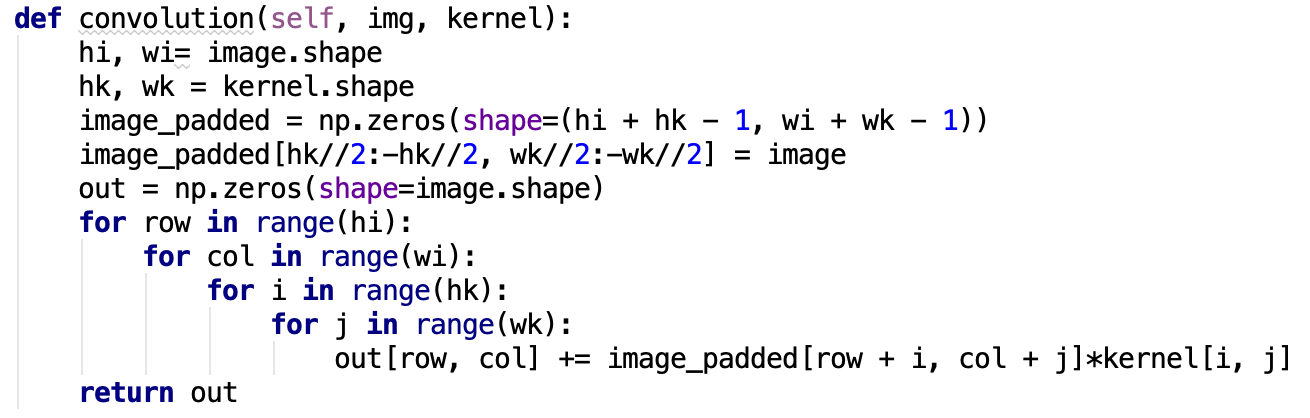
Konvolúcia je matematická operácia pri ktorej každá hodnota vo výstupe je určená ako súčet vstupných hodnôt prenásobených koeficientami váh. Jednotlivé koeficienty váh sú definované pomocou Gaussovho jadra. Konvolúcia je definovaná ako

K(u, v) = (I ⊗ H)(u, v) = ∑ i ∑ j I(u − i, v − j)H(i, j)

kde K(u, v) je hodnota výsledku na pozícii [u, v], symbol ⊗ je operátor konvolúcie, I je vstupný obraz a H je konvolučné jadro.



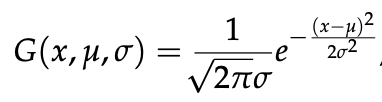
Konvolúciu sme implementovali ako funkciu, kotrá očakáva na vstupe pôvodný orbázok a Gaussove jadro. Následne prechádza obrázok pixel po pixely a prenásobuje jednotlivé hodnoty hodnotamy na zadanom indexe v Gaussovej matici až pokiaľ neprjde cez celý obrázok.



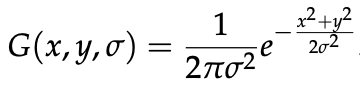
Implementácia konvolúcie

### 2.1.2 Gaussov filter

Medzi lineárne vyhladzovacie filter patrí aj Gaussov filter. Jadro Gaussovho filtra je reprezentované charakteristickým zvonovitým tvarom Gaussovej krivky. Gaussova krivka je v 1D vyjadrená vzťahom

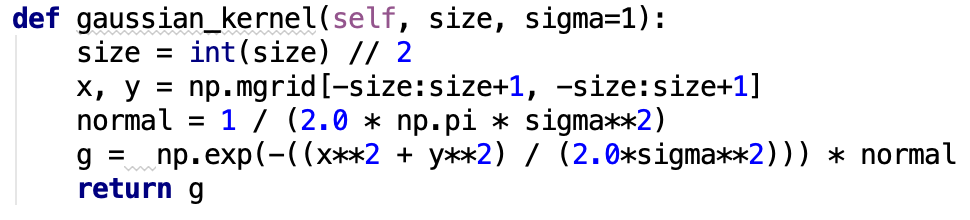
,

kde σ je štandardnou odchýlkou Gaussovho rozloženia a µ je strednou hodnotou. Parameter σ graf rozširuje a parameter µ posúva graf pozdĺž osi x. Podobne Gaussovo rozloženie v 2D, kde µ = 0, je vyjadrené vzťahom

.

Úlohou Gaussovho filtra je hlavne odstránenie šumu a rozmazanie obrazu. Sigma určuje stupeň rozmazania obrazu. Vyššie hodnoty parametru sigma vyžadujú vytvorenie väčšieho jadra, aby bolo reprezentované dostatočne presne. Spravidla veľkosť jadra je určená ako (2K + 1) X (2K + 1), kde K = d3σe. Výstupom Gaussovho filtra je vážený priemer okolia každého pixelu, pričom váhy sú vyššie smerom k stredovým pixelom jadra. Vďaka tejto vlastnosti Gaussov filter poskytuje jemnejšie rozmazanie a zachováva hrany.

Pri implementácii gaussovho jadra sme zadefinovali funckiu “gaussian\_kernel”, ktorá na vstupe očakáva veľkosť matice a sgimu, ktorá je defaultne nastavená na hodnotu 1. Pre výpočet gassovje maticke sme použili vyššie vyjadrený vzťah o Gaussovom rozložení v 2D.



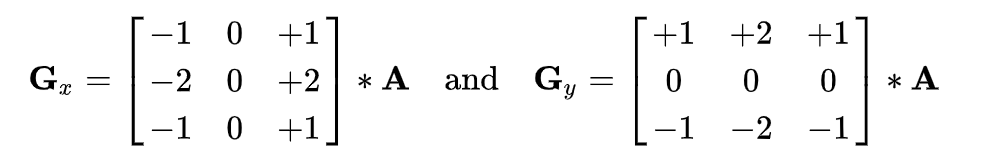
Implementácia Gaussov filter

## 2.1 Gradient calculation – výpočet gradientu

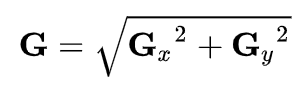
Krok výpočtu gradientu zisťuje intenzitu a smer hrany vypočítaním gradientu obrázka pomocou operátorov detekcie okrajov. Hrany zodpovedajú zmene intenzity pixelov. Najjednoduchším spôsobom na detekciu je použitie filtrov, ktoré zvýrazňujú túto zmenu intenzity v oboch smeroch a to horizontálne na osi x aj vertikálne na osi y.

## 2.1.1 Sobel operator/filter

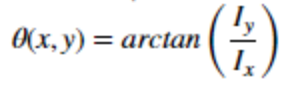
Keď je obraz vyhladený môžeme implementovať konvolúciou s jadrom Sobel Kx a Ky. Sobel operátor alebo filter využíva dve matice Gx pre x os a Gy pre y os o rozmere 3x3 ako konvolucne jadro pri konvolúcii pôvodného orázku.



Súradnica x je tu definovaná ako zvyšujúca sa v smere „doprava“ a súradnica y je definovaná ako zvyšujúca sa v smere „dole“. V každom bode na obrázku sa výsledné aproximácie gradientu môžu kombinovať, aby sa získala veľkosť gradientu, s použitím vzorca



Ďalej môžeme vypočítať aj smer gradientu na to použijeme vzorec



Pri implementácii sme vytvorili konvolučné jadro pre x a y os. Následne sme aplikovali konvolúciu na pôvodný obrázok s vytvoreným konvolučným jadrom pre obe osi zvlášť. Následne sme aplikovali vyššie uvedené vzorce na výpočet veľkosti a smeru gradientu.

