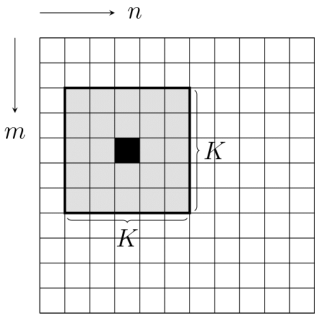
Gaussian filter: acceleration of calculations

A Preprint

In this conspect i am reviewing methods of accelerated approximation of Gaussian filtering calculation. The main idea of Gaussian filtering is smoothing image while reducing noise and preserving important details. From mathematical POV gaussian filter is convolution with Gaussian kernel, on practise finite size of kernels are used. The pixels are weighted depended on the distance from the current pixel.



Gaussian kernel is defined as $g_\sigma[m,n] = \frac{1}{2\pi\sigma^2} e^{-\frac{1}{2}(\frac{m^2+n^2}{\sigma^2})}$

$y[i,j] = \sum_{m=-M}^{M} \sum_{n=-M}^{M} g_\sigma[m,n] \cdot x[i-m, j-n]$

Gaussian kernel is also separable because it can be decomposed in two 1D. There are two main types of approximations: finite impulse response (FIR) filters and infinite impulse response (IIR) filters. FIR allows closer approximations of Gaussian kernel while IIR with their recursive nature are more suitable for situations where computations are more important.

For FIR $y[i] = \sum_{n=i-M}^{i+M} h[i-n] * x[n]$. $h[k]$ - Gaussian weights, cropped on both sides. As for IIR recursive is used. Method is faster as previous outputs are used from memory.

In this work, authors focus on FIR. Stack Blur idea is to use sliding window approach. Edge pixels has a weight 1, which increases by 1 as we move to the center(for example [1, 2, 3, 2, 1]). This reduces time by 4-5 times as previously it was $O(H \cdot W \cdot K^2)$. Now it became $O(H \cdot W \cdot 2K)$. There is also a modification of this algorithm called Bell. It works next way: we store original value and for example quadratic or something like that to imitate this bell form of Gaussian dist.

There is also an approach called Running sums. Authors suggest function from k stairs, height and width of stairs are defined by minimizing MSE. For each row or column cumulative sum is stored, after that window sum is defined by substract particular indices in array.