The Magic Sigma

Dirk Padfield

Senior Research Scientist

Background

- Mobile/embedded devices are everywhere affecting
  - Telecommunications
  - Consumer electronics
  - Household appliances
- Embedded image analysis algorithms include
  - Contrast enhancement
  - Mobile panorama
  - Face recognition
  - Shot steadying
- Robust/fast mobile computing important but challenging

Hardware limitations

- Most embedded devices do not have floating-point operations:
  - Power and heat
  - Speed
  - Precision
- Integer algorithms
  - Use less power because math units less complex
  - Execute faster than floating-point
  - Emulating floats is at least 10x slower
  - Enable added precision - can make every bit count

Integer Gaussian kernels

- Gaussian kernels are often used in machine learning/image/signal processing
  - Anomaly detection - non-outlier (normal) sample distribution
  - Support Vector Machines (SVM) kernels
  - Image smoothing
- Gaussian has several desirable characteristics:
  - Separable
  - Differentiable
  - Result of the central limit theorem
- Discrete Gaussian kernels should have integer values on embedded devices

Motivation

This work is motivated by:

1. The Gaussian kernel is the most ubiquitous kernel
2. A discrete Gaussian kernel should exactly discretize a continuous Gaussian while consisting only of integers
3. Using integers that are all powers-of-two enables bitwise operations and special processing in hardware

The challenge

Here is a puzzler:

- Is it possible to design an integer kernel of arbitrary size whose entries fit a Gaussian function?

The challenge

Here is a puzzler:

- Is it possible to design an integer kernel of arbitrary size whose entries fit a Gaussian function?
- Is it possible to ensure that all the entries are powers-of-two?

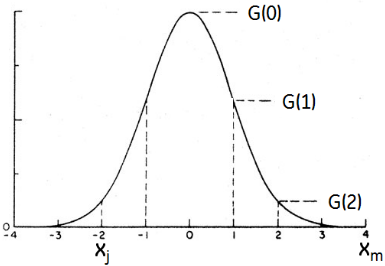
The challenge

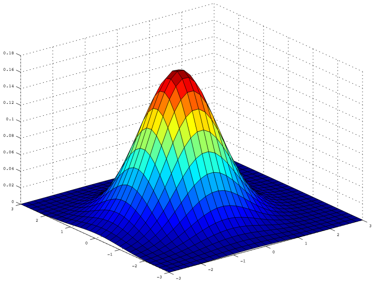
Here is a puzzler:

- Is it possible to design an integer kernel of arbitrary size whose entries fit a Gaussian function?
- Is it possible to ensure that all the entries are powers-of-two?
- Is there a particular sigma (a magic sigma) that meets both of these conditions?

The obligatory Gaussian equation slide

$$G(x;\sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right)$$





Example: is this a Gaussian kernel and what is its σ?

1	2	1
2	4	2
1	2	1

Example: is this a Gaussian kernel and what is its σ?

1	2	1
2	4	2
1	2	1

- Yes, it is a Gaussian kernel.

Example: is this a Gaussian kernel and what is its σ?

1	2	1
2	4	2
1	2	1

- Yes, it is a Gaussian kernel.
- Its $\sigma = \dfrac{1}{\sqrt{\ln 4}} \approx 0.85$

Example: is this a Gaussian kernel and what is its σ?

1	2	1
2	4	2
1	2	1

- Yes, it is a Gaussian kernel.
- Its $\sigma = \dfrac{1}{\sqrt{\ln 4}} \approx 0.85$
- It is a magic kernel because all of its entries are powers-of-two!

That sounds nice, but we must prove it!

$\sqrt{2}$	1	$\sqrt{2}$
1	0	1
$\sqrt{2}$	1	$\sqrt{2}$

Kernel distances

1	2	1
2	4	2
1	2	1

Kernel values

- $G(x;\sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right)$, with $\sigma = \frac{1}{\sqrt{\ln 4}}$

That sounds nice, but we must prove it!

$\sqrt{2}$	1	$\sqrt{2}$
1	0	1
$\sqrt{2}$	1	$\sqrt{2}$

Kernel distances

1	2	1
2	4	2
1	2	1

Kernel values

- $G(x;\sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right)$, with $\sigma = \frac{1}{\sqrt{\ln 4}}$
- $G(0;\sigma) = \sqrt{\frac{\ln(4)}{2\pi}} \exp(0) = \sqrt{\frac{\ln(4)}{2\pi}} = C$

That sounds nice, but we must prove it!

$\sqrt{2}$	1	$\sqrt{2}$
1	0	1
$\sqrt{2}$	1	$\sqrt{2}$

Kernel distances

1	2	1
2	4	2
1	2	1

Kernel values

- $G(x;\sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right)$, with $\sigma = \frac{1}{\sqrt{\ln 4}}$
- $G(0;\sigma) = \sqrt{\frac{\ln(4)}{2\pi}} \exp(0) = \sqrt{\frac{\ln(4)}{2\pi}} = C$
- $G(1;\sigma) = C \exp\left(-\frac{\ln(4)}{2}\right) = C \exp(\ln(1/2)) = C/2$

That sounds nice, but we must prove it!

$\sqrt{2}$	1	$\sqrt{2}$
1	0	1
$\sqrt{2}$	1	$\sqrt{2}$

Kernel distances

1	2	1
2	4	2
1	2	1

Kernel values

- $G(x;\sigma) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{x^2}{2\sigma^2}\right)$, with $\sigma = \frac{1}{\sqrt{\ln 4}}$
- $G(0;\sigma) = \sqrt{\frac{\ln(4)}{2\pi}} \exp(0) = \sqrt{\frac{\ln(4)}{2\pi}} = C$
- $G(1;\sigma) = C \exp\left(-\frac{\ln(4)}{2}\right) = C \exp(\ln(1/2)) = C/2$
- $G(\sqrt{2};\sigma) = C \exp\left(-\frac{\sqrt{2}^2 \ln(4)}{2}\right) = C \exp(\ln(1/4)) = C/4$

That sounds nice, but we must prove it!

$\sqrt{2}$	1	$\sqrt{2}$
1	0	1
$\sqrt{2}$	1	$\sqrt{2}$

Kernel distances

1	2	1
2	4	2
1	2	1

Kernel values

C/4	C/2	C/4
C/2	C	C/2
C/4	C/2	C/4

Scaled kernel values

- $G(x;\sigma) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{x^2}{2\sigma^2}\right)$, with $\sigma = \frac{1}{\sqrt{\ln 4}}$
- $G(0;\sigma) = \sqrt{\frac{\ln(4)}{2\pi}} \exp(0) = \sqrt{\frac{\ln(4)}{2\pi}} = C$
- $G(1;\sigma) = C \exp\left(-\frac{\ln(4)}{2}\right) = C \exp(\ln(1/2)) = C/2$
- $G(\sqrt{2};\sigma) = C \exp\left(-\frac{\sqrt{2}^2 \ln(4)}{2}\right) = C \exp(\ln(1/4)) = C/4$

Okay fine, but...

This is all well and good, but you **told** me the sigma. Maybe you just got lucky.

- By the way, how did you find this σ?
- Does this same σ hold for any size kernel?
- Are there more general σ values that result in Gaussian kernels that do not all have powers-of-two?

Contributions

1. We show how to combine integer arithmetic and bitwise shifts with integer Gaussian kernels
2. We prove that there exists a set of “magic sigmas” that yield integer kernels whose entries are all powers-of-two
3. We prove that the smallest sum “magic sigma” is $\frac{1}{\sqrt{\ln(4)}}$
4. To enable flexible sigma values, we designed an integer kernels algorithm for any arbitrary sigma and kernel size

Previous methods

Three main common approaches to integer kernel generation:

- Rounding method
- Binomial method
- Convolution method

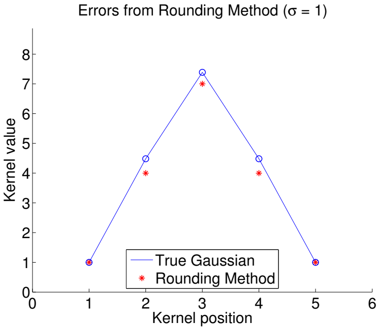
Limitations:

- Result in approximations of the Gaussian kernel
- Do not necessarily result in integer values that are powers-of-two

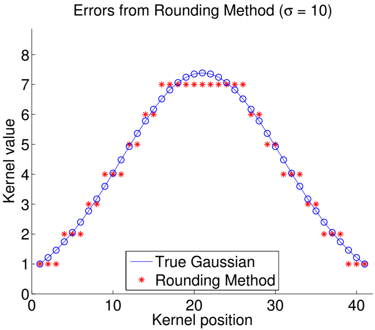
Thus, a more systematic analysis of the kernel construction is necessary in order to derive exact integer kernels

Method 1: Rounding method

1. Create a floating-point Gaussian kernel
2. Normalize the values to make its smallest entry 1
3. Round the entries and store them as integers
4. Divide the kernel by the kernel sum



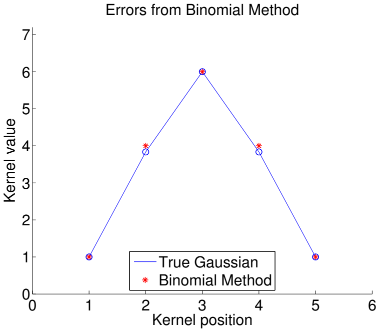
$\sigma = 1, r = 2\sigma$



$\sigma = 10, r = 2\sigma$

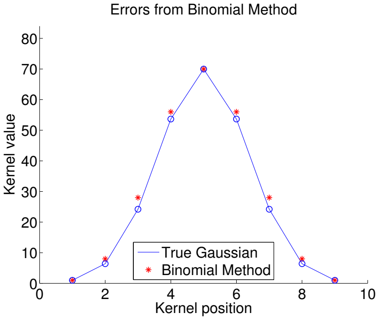
Method 2: Binomial method

- Use odd-length rows of Pascal's triangle: [1 2 1], [1 4 6 4 1]
- Does not enable control over σ for a particular kernel size
- The σ increases with increasing kernel size



[1 4 6 4 1]

$\sigma = 1.0565$



[1 8 28 56 70 56 28 8 1]

$\sigma = 1.3723$

Method 3: Convolution method

- Convolve smaller integer kernels with each other
- Enables the generation of a range of σ values depending on the number of convolutions
- Based on the relationships
  - $G_\sigma = G_{\sigma_1} * G_{\sigma_2}$
  - $\sigma^2 = \sigma_1^2 + \sigma_2^2$.
- Approximation errors increase with larger kernels because of convolution of truncated smaller kernels
- When kernels are chosen from the odd rows of Pascal's triangle, the resulting kernel will be another odd row in Pascal's triangle

Solution requirements

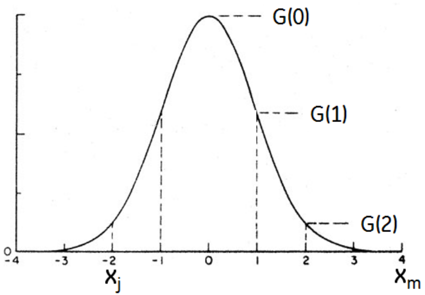
Power-of-two Gaussian kernels require the following constraints:

1. Consistent sigma
2. Integer kernel values
3. Kernel values are powers-of-two

We will demonstrate how to ensure all of these conditions

Notation

- x_j: the distance from the center to a location in the kernel
- x_m: the farthest distance from the center in the kernel
- $G(x_j)$: the value of the Gaussian evaluated at x_j
- $R_{x_i,x_j} = \dfrac{G(x_i)}{G(x_j)}$: the ratio of the Gaussian values at two locations in the kernel
- σ_{x_i,x_j}: the Gaussian σ found using points x_i and x_j



What is meant by "consistent" σ?

- Given a desired σ, the relationship of every pair of points should conform to this value.
- Since Gaussian functions vary by distance, we look at the distances in a kernel.

$\sqrt{2}$	1	$\sqrt{2}$
1	0	1
$\sqrt{2}$	1	$\sqrt{2}$

Distances for 3×3 kernel.

$2\sqrt{2}$	$\sqrt{5}$	2	$\sqrt{5}$	$2\sqrt{2}$
$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$
2	1	0	1	2
$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$
$2\sqrt{2}$	$\sqrt{5}$	2	$\sqrt{5}$	$2\sqrt{2}$

Distances for 5×5 kernel.

Ratios of Gaussians

Any two points x_i,x_j are needed to define the σ of a Gaussian.

$$G(x;\sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right)$$

Taking their ratio makes the points scale invariant.

$$R_{x_i,x_j} = \frac{G(x_i)}{G(x_j)} = \frac{\exp\left(-\frac{x_i^2}{2\sigma^2}\right)}{\exp\left(-\frac{x_j^2}{2\sigma^2}\right)} = \exp\left(\frac{x_j^2 - x_i^2}{2\sigma^2}\right)$$

Solving for σ leads to

$$\sigma_{x_i,x_j} = \sqrt{\frac{x_j^2 - x_i^2}{2\ln(R_{x_i,x_j})}}$$

Solving for σ

We can simplify R_{x_i,x_j} by using $G(0)$ as a reference point.

$$R_{0,x_j} = \frac{G(0)}{G(x_j)} = \exp\left(\frac{x_j^2}{2\sigma^2}\right)$$
$$R_{0,1} = \frac{G(0)}{G(1)} = \exp\left(\frac{1}{2\sigma^2}\right)$$

Using R_{0,x_j}, we can solve for σ_{0,x_j}

Consistent σ

$$\sigma_{0,x_j} = \sqrt{\frac{x_j^2}{2\ln(R_{0,x_j})}}$$

Solving for σ

We can simplify R_{x_i,x_j} by using $G(0)$ as a reference point.

$$R_{0,x_j} = \frac{G(0)}{G(x_j)} = \exp\left(\frac{x_j^2}{2\sigma^2}\right)$$

$$R_{0,1} = \frac{G(0)}{G(1)} = \exp\left(\frac{1}{2\sigma^2}\right)$$

Using R_{0,x_j}, we can solve for σ_{0,x_j}

Consistent σ

$$\sigma_{0,x_j} = \sqrt{\frac{x_j^2}{2\ln(R_{0,x_j})}} = \sqrt{\frac{x_j^2}{2\ln\left((R_{0,1})^{x_j^2}\right)}}$$

Solving for σ

We can simplify R_{x_i,x_j} by using $G(0)$ as a reference point.

$$R_{0,x_j} = \frac{G(0)}{G(x_j)} = \exp\left(\frac{x_j^2}{2\sigma^2}\right)$$

$$R_{0,1} = \frac{G(0)}{G(1)} = \exp\left(\frac{1}{2\sigma^2}\right)$$

Using R_{0,x_j}, we can solve for σ_{0,x_j}

Consistent σ

$$\sigma_{0,x_j} = \sqrt{\frac{x_j^2}{2\ln(R_{0,x_j})}} = \sqrt{\frac{x_j^2}{2\ln\left(\left(R_{0,1}\right)^{x_j^2}\right)}} = \sqrt{\frac{1}{2\ln\left(R_{0,1}\right)}}$$

Thus, when $R_{0,x_j} = \left(R_{0,1}\right)^{x_j^2}$ for all x_j, σ_{0,x_j} is "consistent" because it does not depend on x_j.

Ensuring Integer Values

Next, we need to ensure that all values are integers.

- We can find the value at any location x_j in the kernel $G(x_j)$ relative to the value at the center $G(0)$.
- $G(x_j) = \frac{G(0)}{R_{0,x_j}} = \frac{G(0)}{(R_{0,1})^{x_j^2}}$
- By definition, the smallest value in the kernel $G(x_m)$ will be at the farthest distance x_m from the center.

Ensuring integer values

If $R_{0,1} = \frac{p}{q}$ is rational and $G(0) = (p)^{x_m^2}$, all values in the kernel will be integers.

- This can easily be proven (see proof in paper).

Ensuring powers-of-two: the "magic sigma" family

- Powers-of-two enable efficient bit-wise shifts
- Recall that $R_{0,x_j} = (R_{0,1})^{x_j^2}$
- If $R_{0,1}$ is a power-of-two, then all kernel entries are powers-of-two because x_j^2 is always an integer.
- Thus, $R_{0,1} = 2^i$ (where $i > 0$ is an integer) leads to the "magic sigmas"

Ensuring powers-of-two: the “magic sigma” family

- Powers-of-two enable efficient bit-wise shifts
- Recall that $R_{0,x_j} = (R_{0,1})^{x_j^2}$
- If $R_{0,1}$ is a power-of-two, then all kernel entries are powers-of-two because x_j^2 is always an integer.
- Thus, $R_{0,1} = 2^i$ (where $i > 0$ is an integer) leads to the “magic sigmas”

The “magic sigma” family

$$\sigma = \frac{1}{\sqrt{2\ln(2^i)}}$$

The ultimate “magic sigma”

We can go one step further by setting $R_{0,1} = 2$

- This yields the smallest kernel values (see proof in paper)
- This also yields the largest σ for a power-of-two Gaussian kernel (see proof in paper)

The ultimate "magic sigma"

We can go one step further by setting $R_{0,1} = 2$

- This yields the smallest kernel values (see proof in paper)
- This also yields the largest σ for a power-of-two Gaussian kernel (see proof in paper)

The "magic sigma"

$$\sigma = \frac{1}{\sqrt{\ln(4)}} \approx 0.849$$

If you've fallen asleep, this slide is worth waking up for...

Ingredients needed to ensure power-of-two Gaussian kernels:

- Consistent σ: $\sigma_{0,x_j} = \dfrac{1}{\sqrt{2\ln\left(R_{0,1}\right)}}$
- Integer constraint: $R_{0,1}$ must be rational.
- Power-of-two: $R_{0,1} = 2^i$, where $i > 0$ is an integer

This leads to the “magic sigma” when i = 1:

$$\sigma = \frac{1}{\sqrt{\ln(4)}} \approx 0.849$$

Enabling greater σ flexibility

Magic sigmas

- Advantage: efficient computation via bitwise shifts
- Disadvantage: limited σ options (max ≈ 0.849)
- Tradeoff: power-of-two kernels vs. flexible σ
- Need algorithm for integer Gaussian kernels for any specified σ

General algorithm for integer Gaussian kernels

Input: Gaussian σ, kernel radius r, and kernel dimension D
Output: Integer Gaussian kernel

1. Calculate real number $R'_{0,1} = \exp\left(\frac{1}{2\sigma^2}\right)$.
2. Since $R_{0,1}$ must be rational, find the "best rational approximation" $R_{0,1} = \frac{p}{q}$ of integers that approximates $R'_{0,1}$ using the "continued fraction"
3. Calculate the Euclidean distance from the kernel center to each kernel entry x_j, where $j = 1...(2r+1)^D$
4. Calculate $G(0) = (p)^{x_m^2}$
5. Find each kernel entry $G(x_j) = \frac{G(0)}{(R_{0,1})^{x_j^2}}$
6. Apply the kernel to the image and divide the result by the kernel sum

Example: 3×3 kernel using "magic sigma" $\sigma = \frac{1}{\sqrt{\ln(4)}}$

1. Calculate $R'_{0,1} = \exp\left(\frac{1}{2\sigma^2}\right) = \exp\left(\frac{\ln(4)}{2}\right) = 2$
2. Find the "best rational approximation" $R_{0,1} = \frac{p}{q} = \frac{2}{1}$
3. Calculate the Euclidean distance from the kernel center to each kernel entry $x_j = 0, 1, \sqrt{2} \rightarrow x_m = \sqrt{2}$
4. Calculate $G(0) = (p)^{x_m^2} = (2)^{(\sqrt{2})^2} = 4$
5. Find each kernel entry $G(x_j) = \dfrac{G(0)}{(R_{0,1})^{x_j^2}}$

   $G(1) = \frac{4}{2^{(1)^2}} = 2$ and $G(\sqrt{2}) = \frac{4}{(2)^{(\sqrt{2})^2}} = 1$
6. Apply the kernel to the image and divide the result by the kernel sum

3×3 Gaussian integer kernels with $R_{0,1}$ set to an integer

1	2	1
2	4	2
1	2	1

$R_{0,1} = 2$

$\sigma \approx 0.85$

1	2	1

1	3	1
3	9	3
1	3	1

$R_{0,1} = 3$

$\sigma \approx 0.67$

1	3	1

1	4	1
4	16	4
1	4	1

$R_{0,1} = 4$

$\sigma \approx 0.60$

1	4	1

3×3 Gaussian integer kernels with $R_{0,1}$ a rational number

4	6	4
6	9	6
4	6	4

$R_{0,1} = 3/2$

$\sigma \approx 1.11$

2	3	2

16	20	16
20	25	20
16	20	16

$R_{0,1} = 5/4$

$\sigma \approx 1.50$

4	5	4

100	110	100
110	121	110
100	110	100

$R_{0,1} = 11/10$

$\sigma \approx 2.29$

10	11	10

5×5 Gaussian "magic sigma" integer kernel

Gaussian kernel values

1	8	16	8	1
8	64	128	64	8
16	128	256	128	16
8	64	128	64	8
1	8	16	8	1

$R_{0,1} = 2$, $\sigma \approx 0.849$

5×5 Gaussian "magic sigma" integer kernel

Gaussian kernel values

1	8	16	8	1
8	64	128	64	8
16	128	256	128	16
8	64	128	64	8
1	8	16	8	1

$R_{0,1} = 2, \sigma \approx 0.849$

Power-of-two exponents

0	3	4	3	0
3	6	7	6	3
4	7	8	7	4
3	6	7	6	3
0	3	4	3	0

$R_{0,1} = 2, \sigma \approx 0.849$

The kernel values can become large for large kernels. Solution: store the exponents, and use them for bitwise shifting

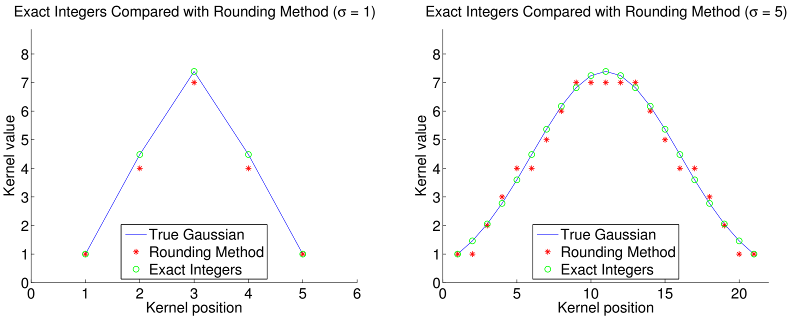
5×5 Gaussian integer kernels with $R_{0,1}$ a rational number

256	864	1296	864	256
864	2916	4374	2916	864
1296	4374	6561	4374	1296
864	2916	4374	2916	864
256	864	1296	864	256

$R_{0,1} = 3/2$, $\sigma \approx 1.1105$

- The kernel values increase with increasing numerator and denominator of $R_{0,1}$ and as the kernel size increases
- Separable 1D filters can instead be used
- If kernel becomes large, Fourier domain can be used

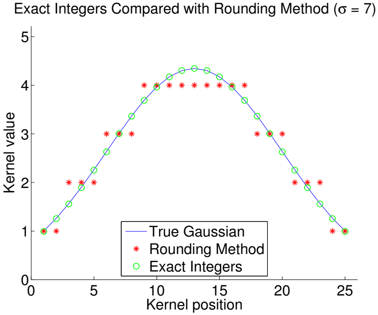
Exact integer results compared with the rounding approach



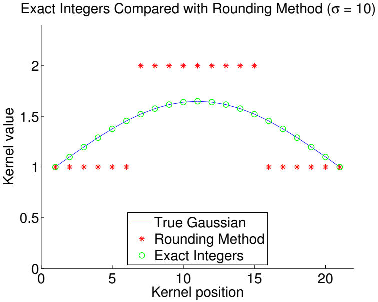
$$\sigma = 1,\ R_{0,1} = \frac{61}{37}$$

$$\sigma = 5,\ R_{0,1} = \frac{101}{99}$$

Exact integer results compared with the basic rounding approach



$$\sigma = 7,\ R_{0,1} = \frac{98}{97}$$



$$\sigma = 10,\ R_{0,1} = \frac{200}{199}$$

Summary: contributions

1. We proved for the first time the existence of a set of “magic sigmas” that generate integer Gaussian kernels whose entries are powers-of-two
2. The magic sigma values are $\sigma = \dfrac{1}{\sqrt{2\ln(2^i)}}$
3. The simplest and largest of these $\sigma = \dfrac{1}{\sqrt{\ln(4)}} \approx 0.849$
4. We developed a flexible algorithm for generating integer Gaussian kernels for arbitrary σ and kernel size.

Discussion points: what does this all mean?

- Integer operations improve power, speed, and precision on mobile devices
- If possible, choose a “magic sigma” rather than an arbitrary number. This enables advantage of integer kernels and bit-wise shifts
- For flexible sigma, use the algorithm to generate integer kernels
- Efficient algorithm implementations can take advantage of these kernels