university of groningen

Image Thresholding

M.H.F. Wilkinson
Bernoulli Institute
University of Groningen

- The Principle
- Histogram-based methods
- The Otsu Method
- RATS and relatives
- Local thresholding

- *image segmentation* means separating an image into *objects* and *background*.
- This is an *extremely difficult* task, mainly because it usually not clear what exactly an *object* is.
- a simple segmentation method uses *thresholding* of the image: pixels above the 'threshold' are declared *object*, those below the threshold *background*.

A binary image consists of *1-pixels* (pixels with value 1, foreground) and *0-pixels* (pixels with value 0, background).

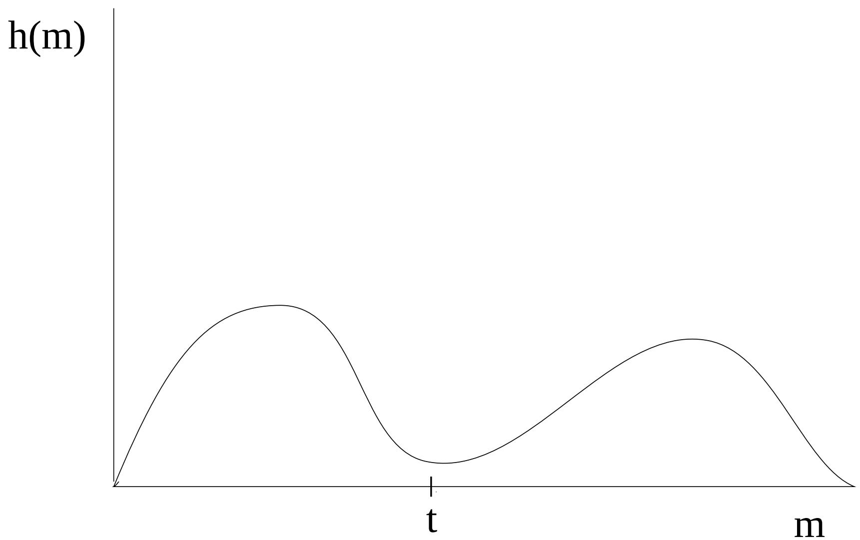
Let $I(x, y)$ be a grey value image. Let t be a fixed grey value called the *threshold*.
Define a binary image I_t, called the thresholded image, by:

$$I_t(x, y) := 1 \quad if \quad I(x, y) \geq t$$
$$I_t(x, y) := 0 \quad if \quad I(x, y) < t$$

(a) input image.

(b) threshold image of (a)

- if the histogram is *bimodal*, i.e has two peaks, the threshold is simple to choose *globally*: choose t somewhere in the valley between the two maxima.
- For more complex images a *local* thresholding may be necessary.



- Often the valley in the histogram is not deep enough
- It can be made deeper by removing *high gradient* pixels
- These edge pixels should be rare in the image, and therefore lie in the valley.
- Alternatively, you can reject *low gradient* pixels, and use the mode of the resulting histogram as threshold

Kapur et al's Method

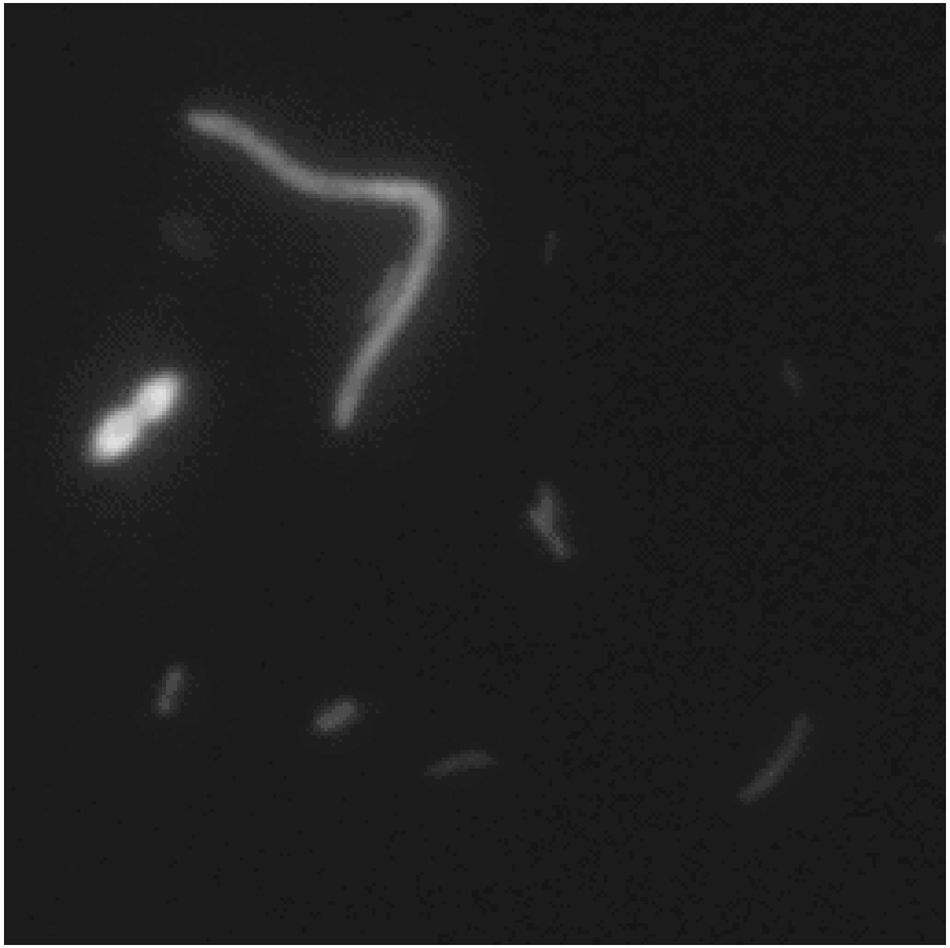
- Note that

$$\omega(t) = \sum_{i=0}^{t} p_i \tag{1}$$

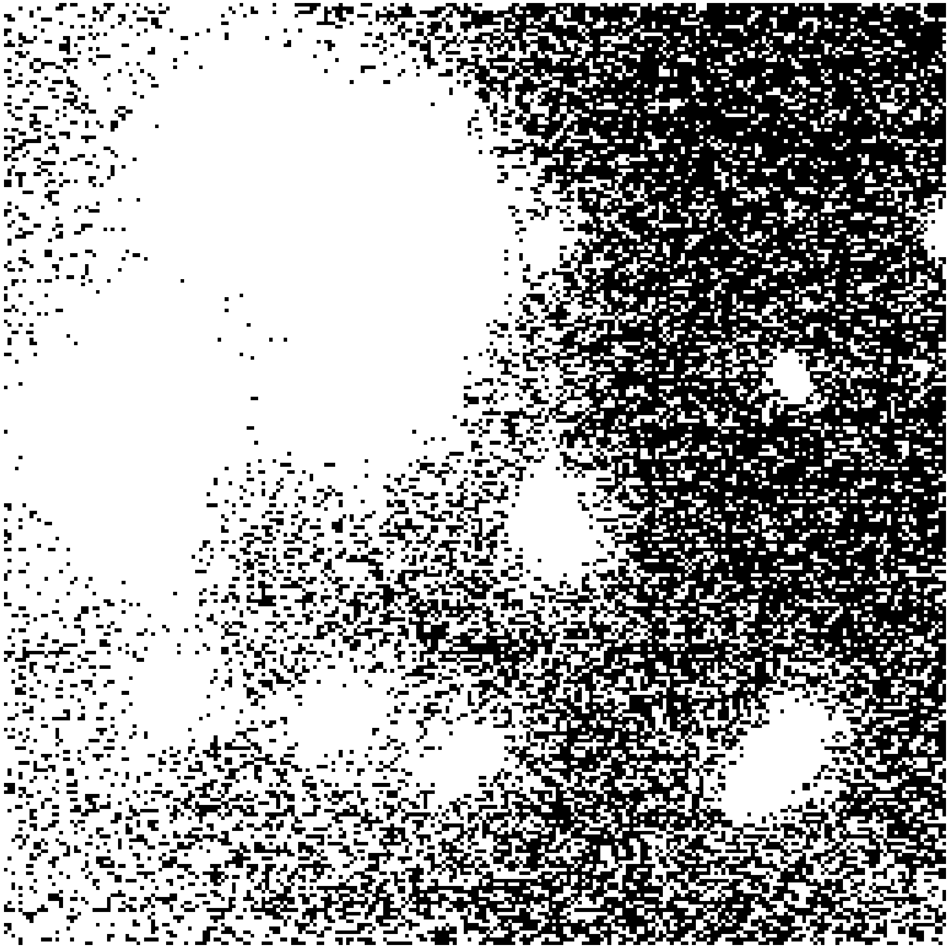
- We now loop through all possible thresholds
- Compute the sum of $H_{C_1}(t)$ and $H_{C_2}(t)$
- Select the threshold which maximizes this statistic, i.e.,

$$T_{\text{opt}} = \arg\max(H_{C_1}(t) + H_{C_2}(t)) \tag{2}$$

Kapur et al's Method



original



thresholded

- The method by Otsu (1979) is a good general purpose thresholding method.
- The basic idea is that we should minimize the within-class variances σ_1^2 and σ_2^2:

$$\sigma_1^2 = \sum_{i=0}^{t}(i - \mu_1)^2 \frac{p_i}{\omega(t)} \tag{3}$$

and

$$\sigma_2^2 = \sum_{i=t+1}^{T_{\max}}(i - \mu_2)^2 \frac{p_i}{1 - \omega(t)} \tag{4}$$

- In which

$$\mu_1 = \sum_{i=0}^{t} i \frac{p_i}{\omega(t)} \tag{5}$$

and

$$\mu_2 = \sum_{i=t+1}^{T_{\max}} i \frac{p_i}{1-\omega(t)} \tag{6}$$

- In principle we can compute these for all thresholds, but this is costly

- We observe that the total variance σ^2 is the sum of the within-class variances and the between-class variance σ_B^2

$$\sigma^2 = \sigma_B^2 + \omega(t)\sigma_1^2 + (1 - \omega(t))\sigma_2^2 \tag{7}$$

- Minimizing within-class variance therefore equals maximizing between-class variance
- The statistic σ_B^2 is computed from

$$\sigma_B^2(T) = \omega(t)(\mu_1(t) - \mu)^2 + (1 - \omega(t))(\mu_2(t) - \mu)^2 \tag{8}$$

with μ the mean grey level of the image

- We can write μ as

$$\mu = \omega(t)\mu_1(t) + (1 - \omega(t))\mu_2(t) \tag{9}$$

- This leads to

$$\sigma_B^2(T) = \omega(t)(1 - \omega(t))(\mu_1(t) - \mu_2(t))^2 \tag{10}$$

- The desired threshold is simply

$$T_{\text{opt}} = \arg\max \sigma_B^2(T) \tag{11}$$

Otsu's Method V

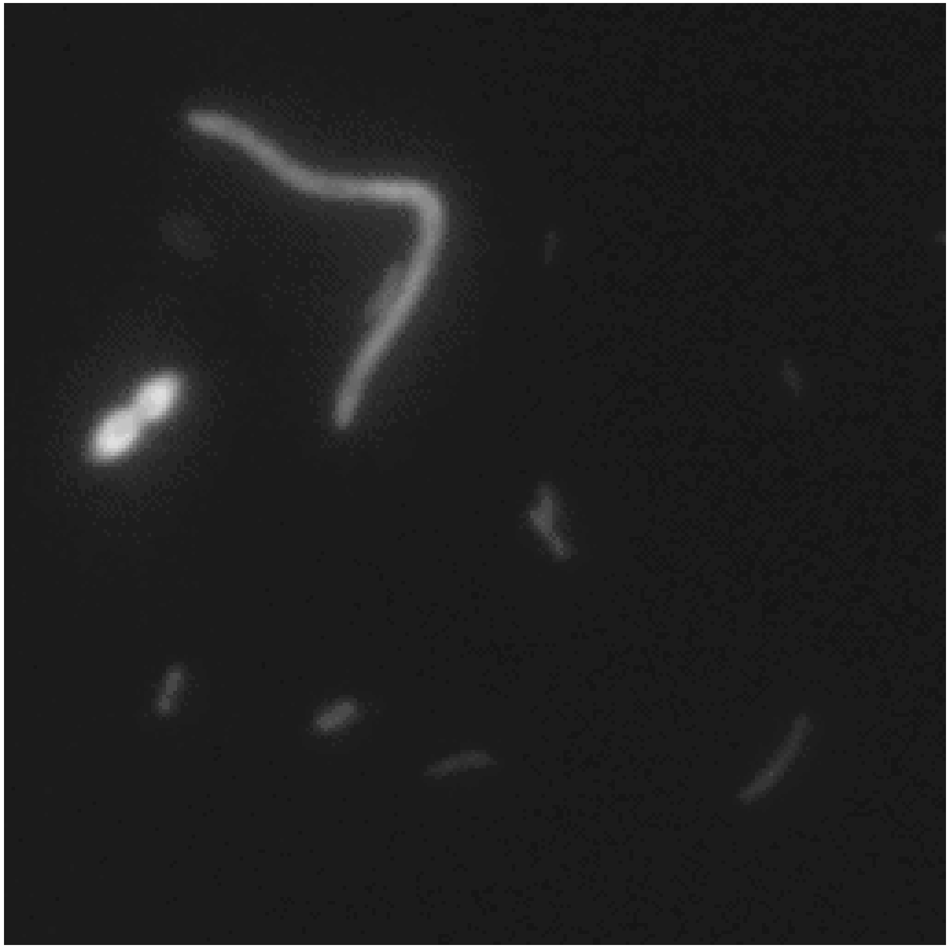
- Again, computing this for every threshold is expensive
- To speed things up, we can create recursive equations for μ_1 and μ_2

$$\mu_1(t+1) = \frac{\omega(t)\mu_1(t) + (t+1)p_{t+1}}{\omega(t+1)} \tag{12}$$

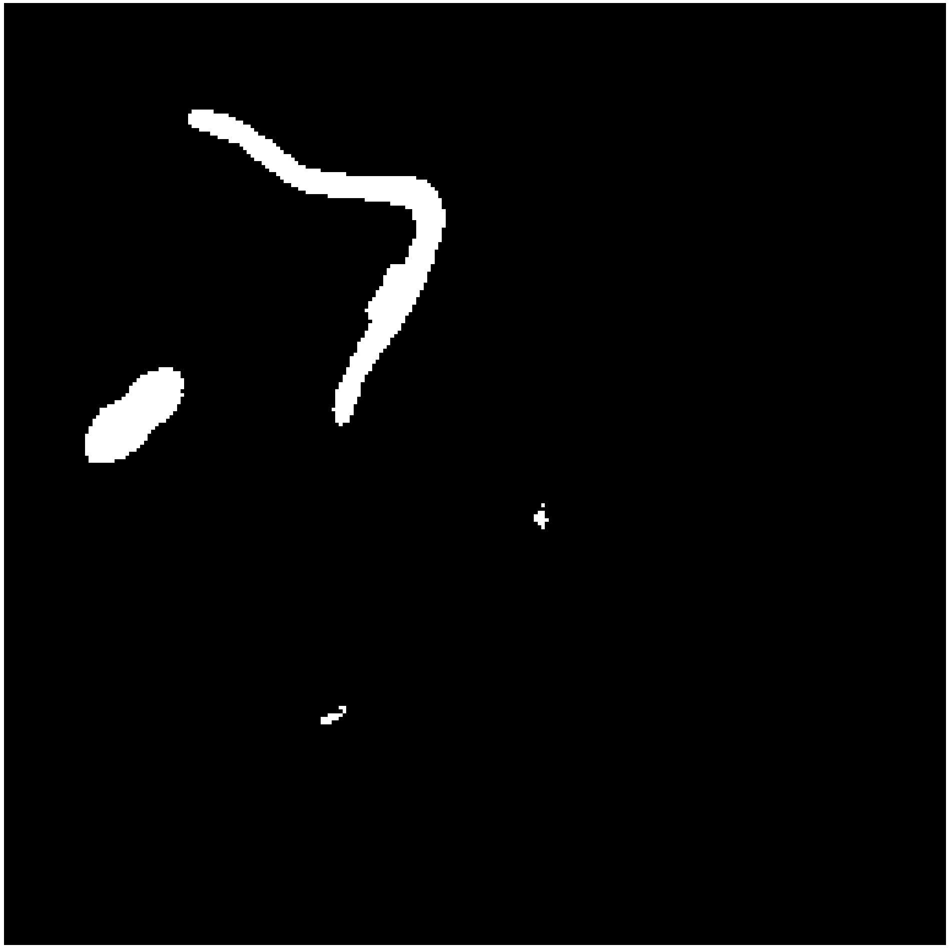
and

$$\mu_2(t+1) = \frac{\mu - \omega(t+1)\mu_1(t+1)}{1 - \omega(t+1)} \tag{13}$$

- This allows rapid computation os σ_B^2 for all thresholds



original



thresholded

- RATS is a simple and fast method for bilevel thresholding of grey scale images.
- Kittler et al. (1985) show that the optimal threshold T in a noise-free image is given by

$$T = \frac{\sum e(x,y)p(x,y)}{\sum e(x,y)}, \tag{14}$$

in which $p(x,y)$ is the grey level at (x,y) and the edge strength $e(x,y)$

- In the original algorithm $e(x, y)$ is given by

$$e(x, y) = \max(g_x(x, y), g_y(x, y)), \tag{15}$$

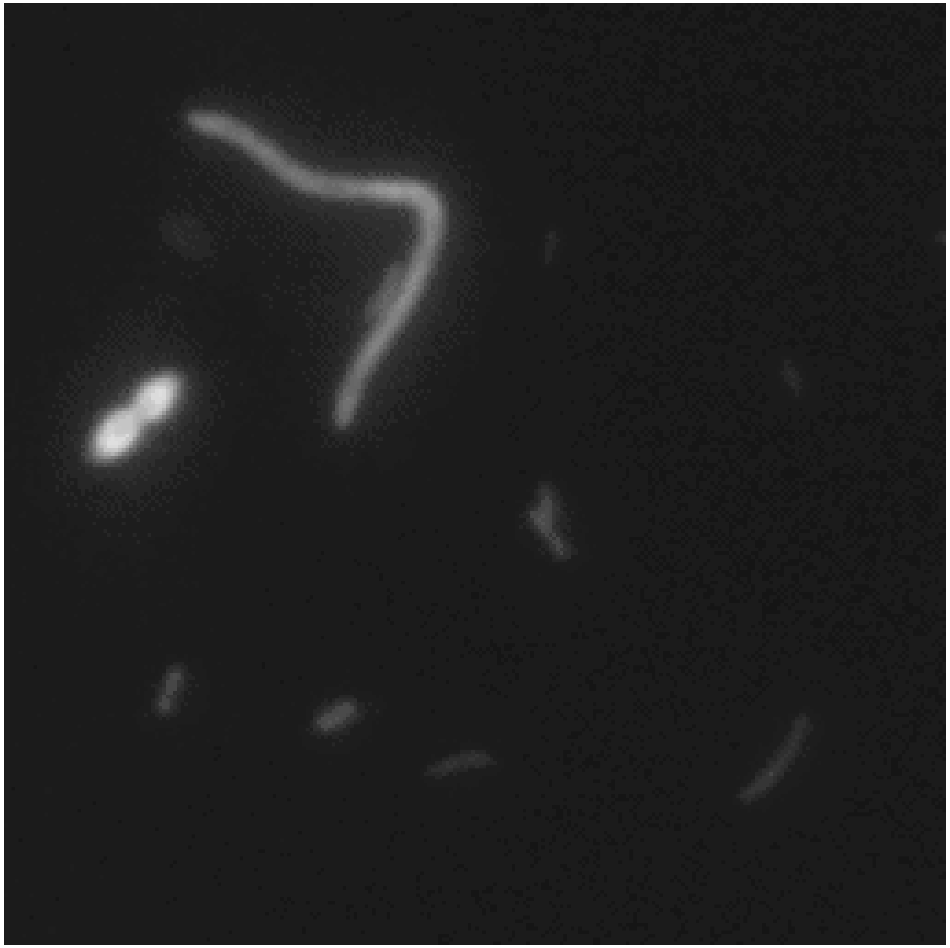
with

$$g_x(x, y) = |p(x-1, y) - p(x+1, y)| \tag{16}$$

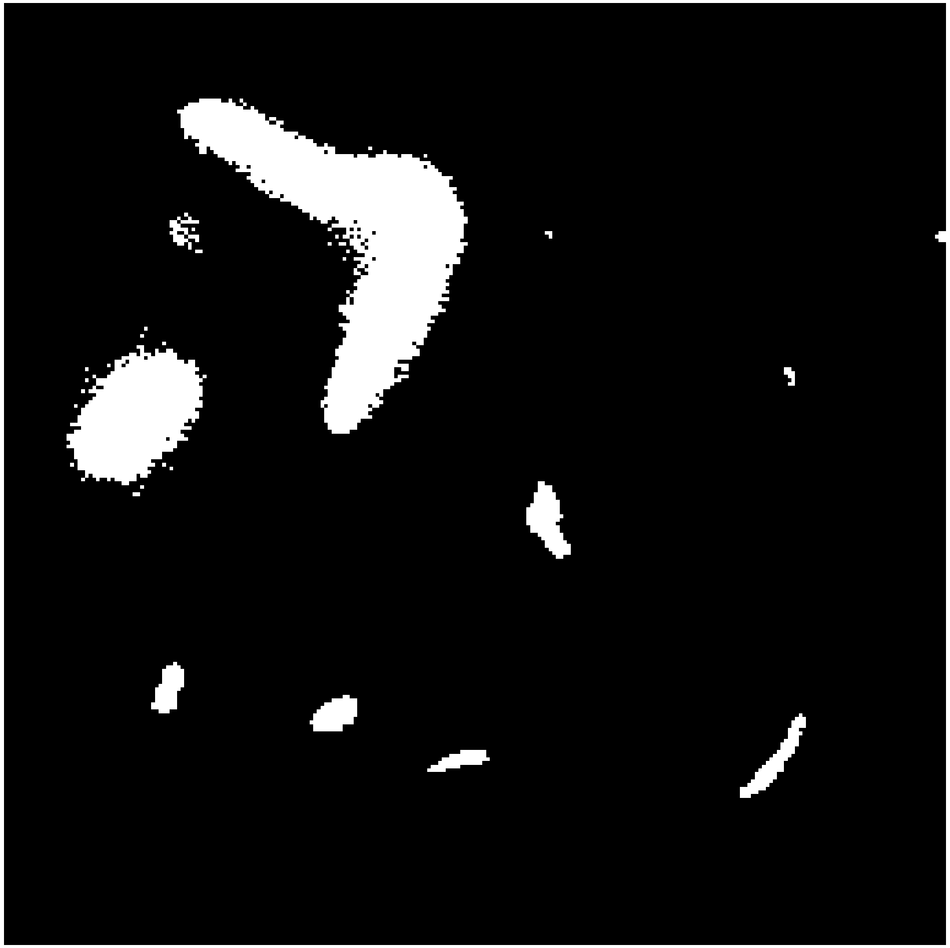
and

$$g_y(x, y) = |p(x, y-1) - p(x, y+1)|. \tag{17}$$

RATS Results



original



thresholded

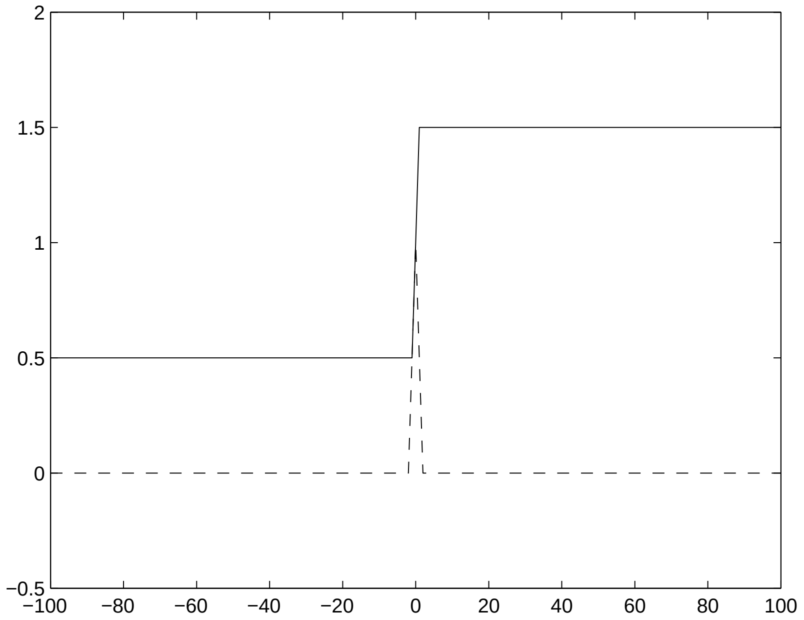
- Consider an ideal step $S_A(x)$ at $x = 0$ of amplitude A on a constant background B.
- This function can be split into an even and an odd part,

$$S_A(x) = B + \frac{A}{2} + \frac{A}{2}\operatorname{sgn}(x) \tag{18}$$

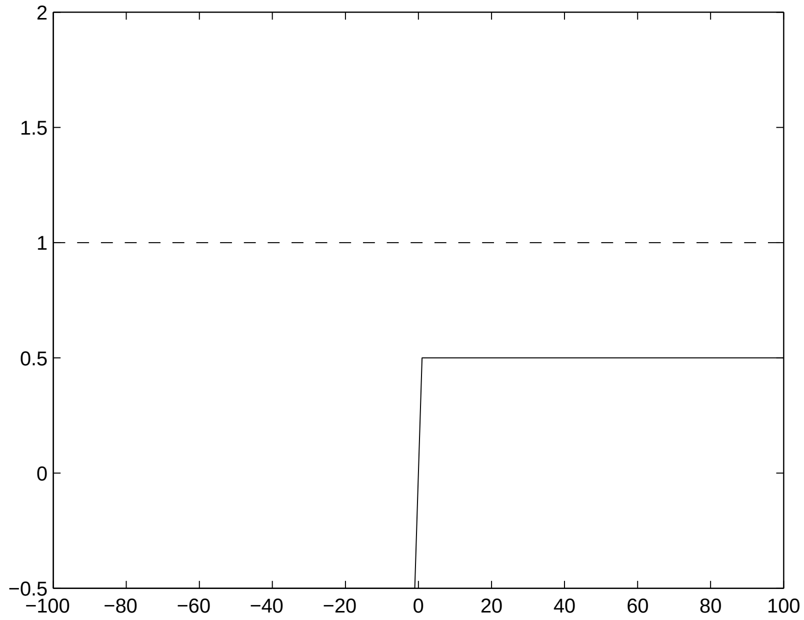
with $\operatorname{sgn}(x)$ the sign of x.

- Let $S_A(x)$ be convolved with an even point-spread-function (PSF) $k(x)$ with a unity integral over $(-\infty, \infty)$.
- The result $S_A * k(x)$ can also be written as an even and odd part,

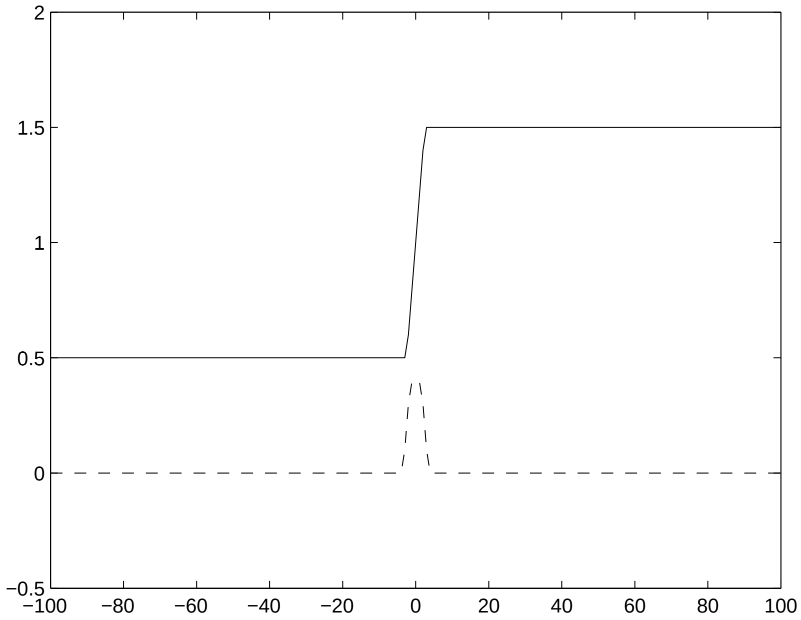
$$S_A * k(x) = B + \frac{A}{2} + \frac{A}{2} k * \operatorname{sgn}(x) \tag{19}$$



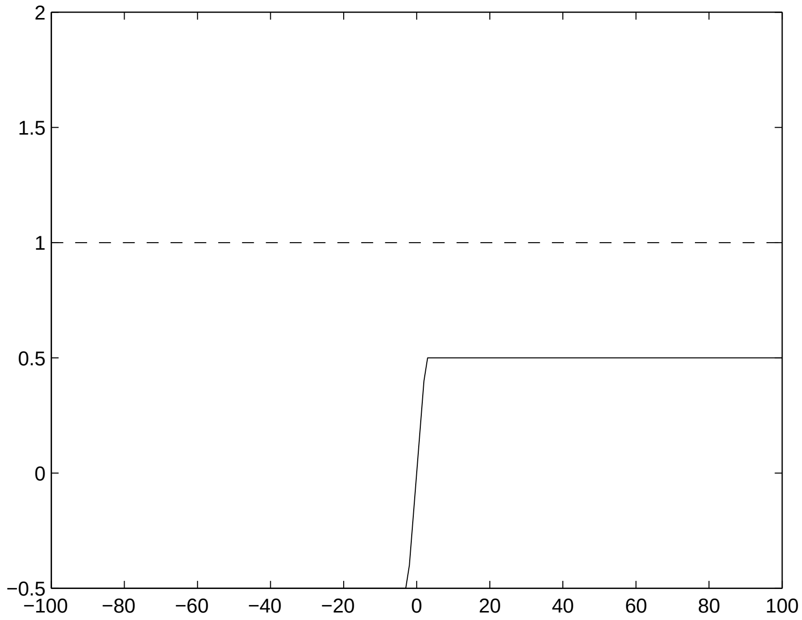
A step and its gradient



Even and odd parts



Blurred step and gradient



Even and odd parts

- We now simulate an edge detector by a weight function $w(x)$
- $w(x)$ must be even, is zero outside $[-a, a]$ for some finite a, and which has a nonzero integral over $[-a, a]$.
- We now model T as follows

$$\begin{aligned} T &= \frac{\int_{-a}^{a} S_A * k(x) w(x)}{\int_{-a}^{a} w(x)} \\ &= \frac{\frac{A}{2} \int_{-a}^{a} k(x) * \operatorname{sgn}(x) w(x) + (B + \frac{A}{2}) \int_{-a}^{a} w(x)}{\int_{-a}^{a} w(x)} \\ &= B + \frac{A}{2} \end{aligned} \tag{20}$$

- This is the ideal threshold, irrespective of k
- Any edge detector with *even* response to a step edge is usable
- This rules out Laplacian and LoG (Marr-Hildreth) filters
- For rotational symmetry reasons

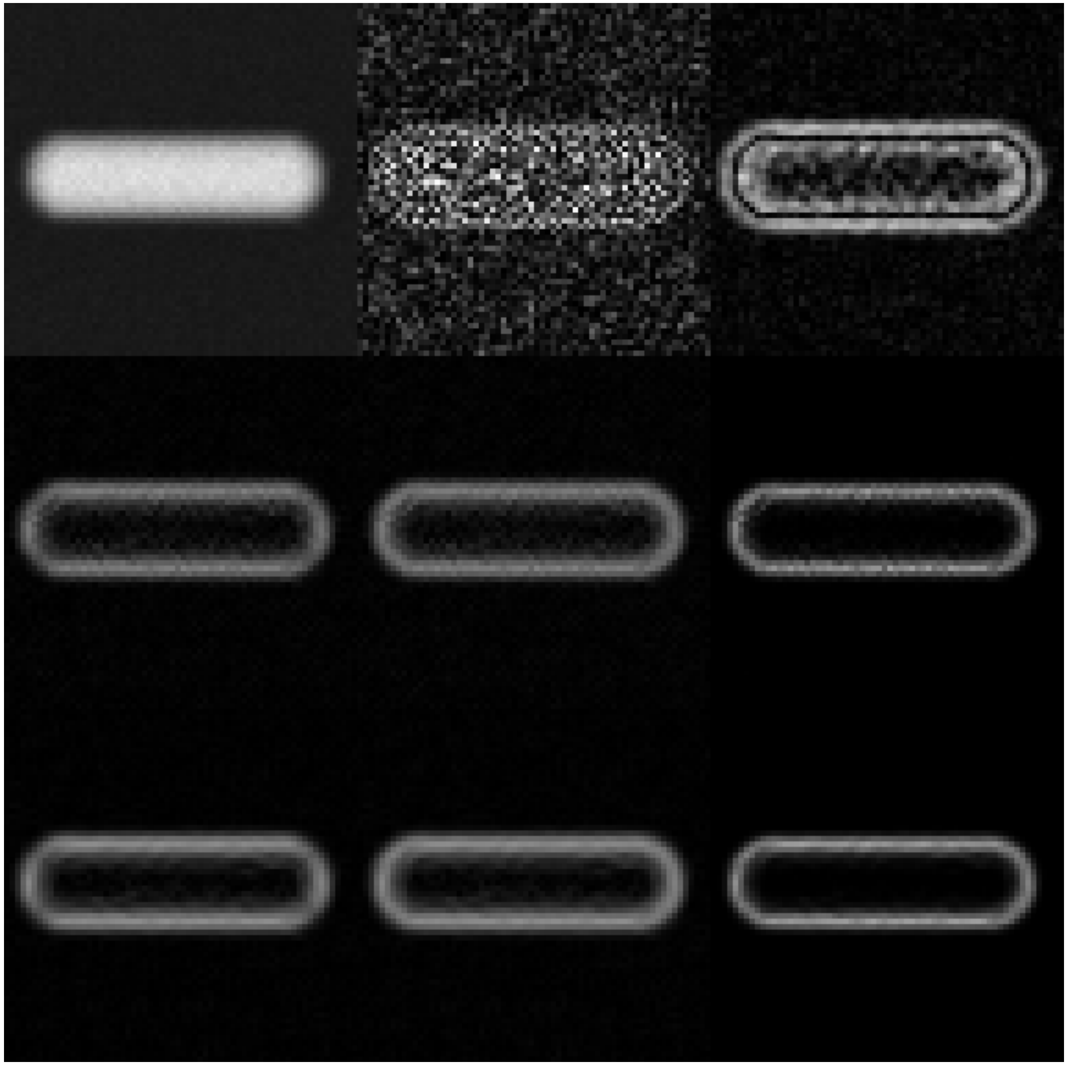
$$g(x, y) = \sqrt{g_x^2(x, y) + g_y^2(x, y)} \tag{21}$$

and

$$g^2(x, y) = g_x^2(x, y) + g_y^2(x, y) \tag{22}$$

are very suitable

Which edge detector to use?



- The previous 1-D result transfers to any straight edge in any number of dimensions
- Consider the following edge detector

$$w_r(x,y) = \begin{cases} 1 & \text{if } (x,y) \text{ is within } r \text{ of an edge} \\ 0 & \text{otherwise} \end{cases} \tag{23}$$

- If the object of interest is a circle of radius R we have

$$T = B + \frac{A(\pi R^2 - \pi(R-r)^2)}{\pi(R+r)^2 - \pi(R-r)^2} = B - \frac{A}{2}\left(1 - \frac{r}{2R}\right) \tag{24}$$

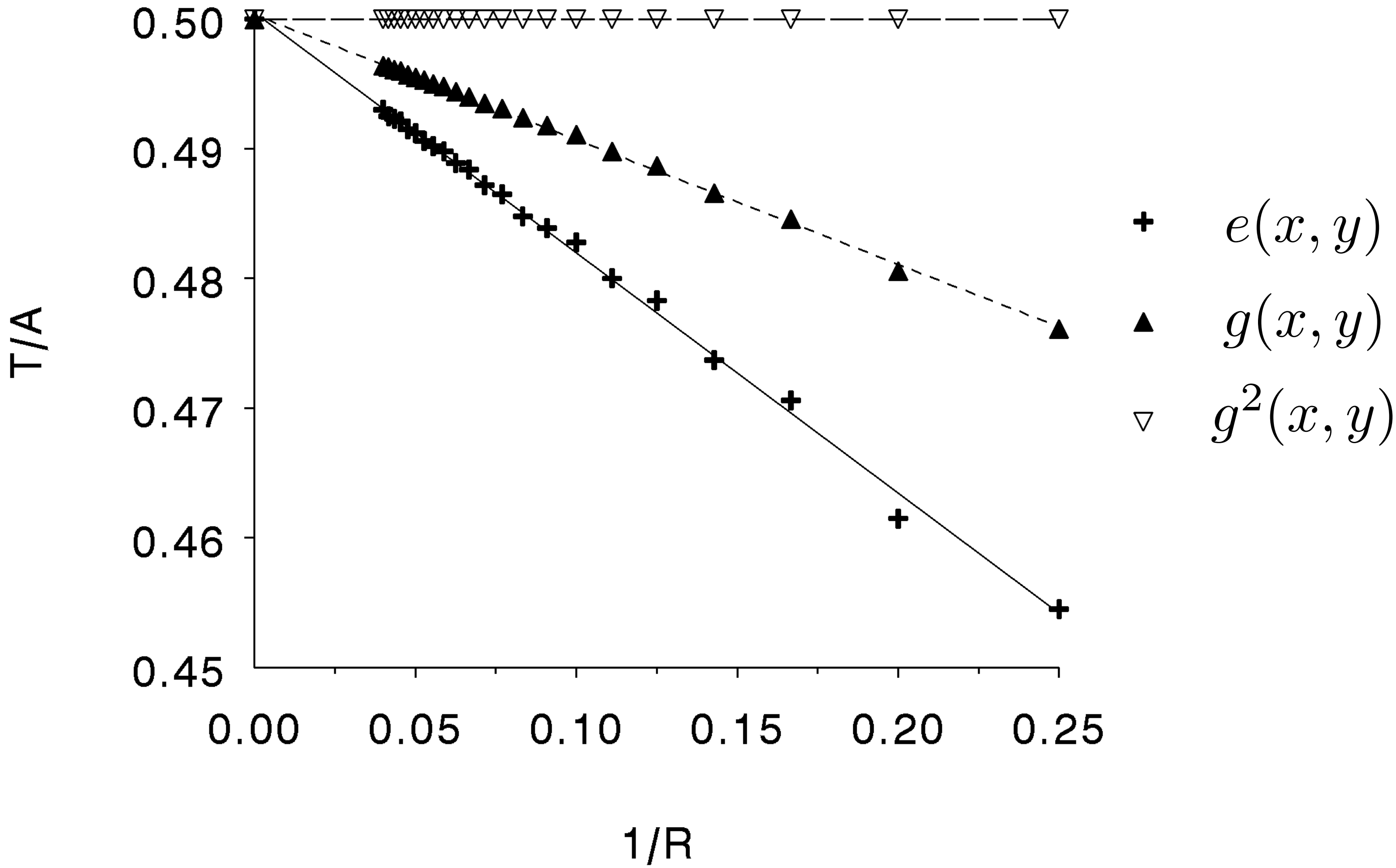
- The bias also exists in 3-D. For spheres of radius R we have

$$T = B + \frac{A(\pi R^3 - \pi(R-r)^3)}{\pi(R+r)^3 - \pi(R-r)^3} = B - \frac{A}{2}\left(1 - \frac{3Rr}{6R^2 + 2r^2}\right) \quad (25)$$

- The curvature bias is caused by the fact that the surface area in the outer curve is larger than that in the inner
- The bias vanishes if

$$r \ll R \quad (26)$$

Measuring Curvature Bias



Noise Bias

- Kittler et al (1985) already studied the behaviour of the statistic with Gaussian noise of standard deviation σ.
- Let the fraction of foreground pixels be q, and the mean respons to noise of $e(x, y)$ be $\nu\sigma$
- The statistic then becomes

$$T_\sigma = \frac{q(B + A)\nu\sigma + (1 - q)B\nu\sigma}{\nu\sigma} = B + \frac{A}{2} + \left(q - \frac{1}{2}\right)A \qquad (27)$$

- Note that this is the *low* S/N situation

- In *high* S/N images we have

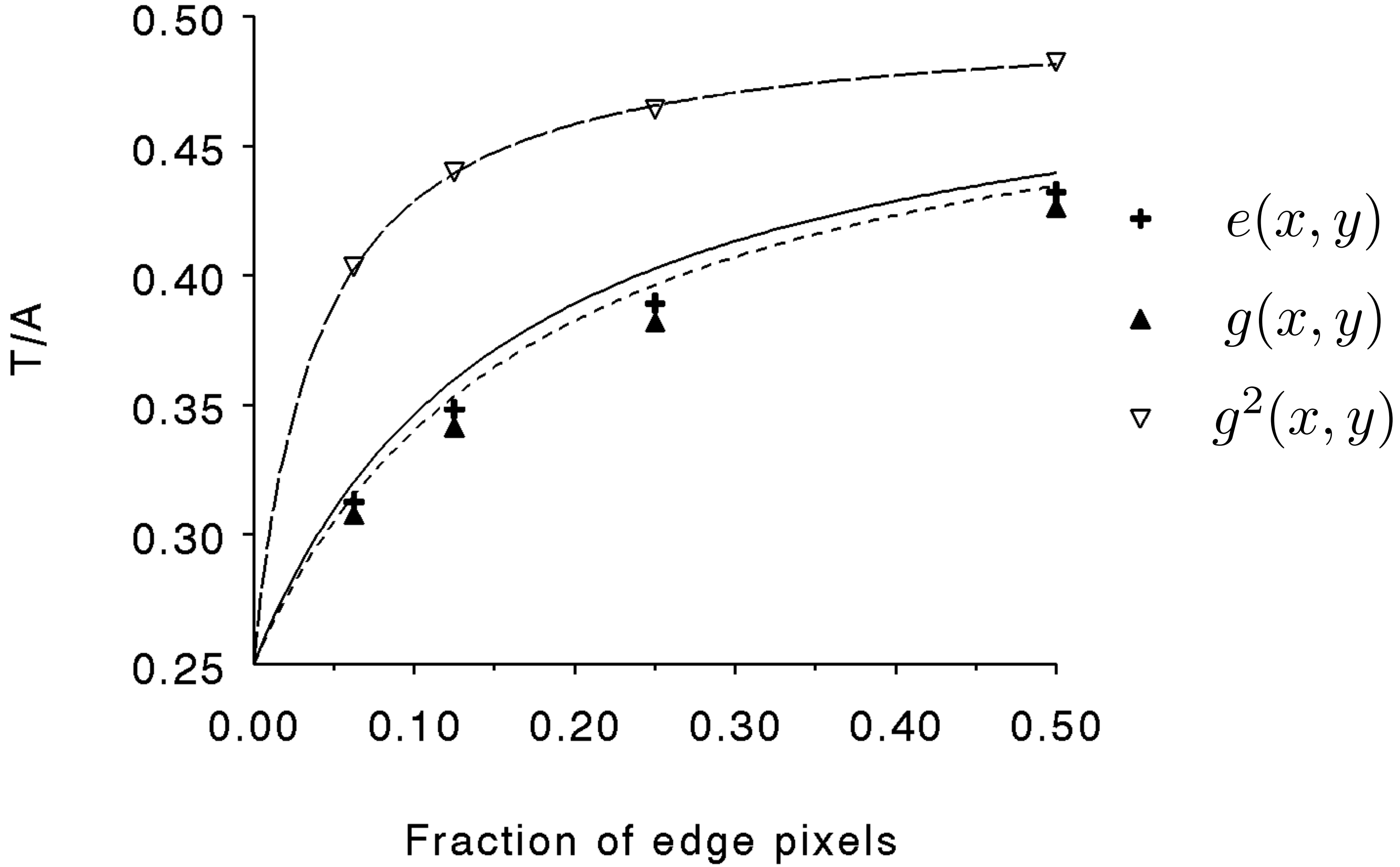
$$T = \frac{W_{\text{edge}}(B + \frac{A}{2}) + W_{\text{noise}}(B + qA)}{W_{\text{edge}} + W_{\text{noise}}} \tag{28}$$

- Let the fraction of edge pixels be ϵ, and $g|A|$ the response to edges of amplitude A
- The statistic then becomes

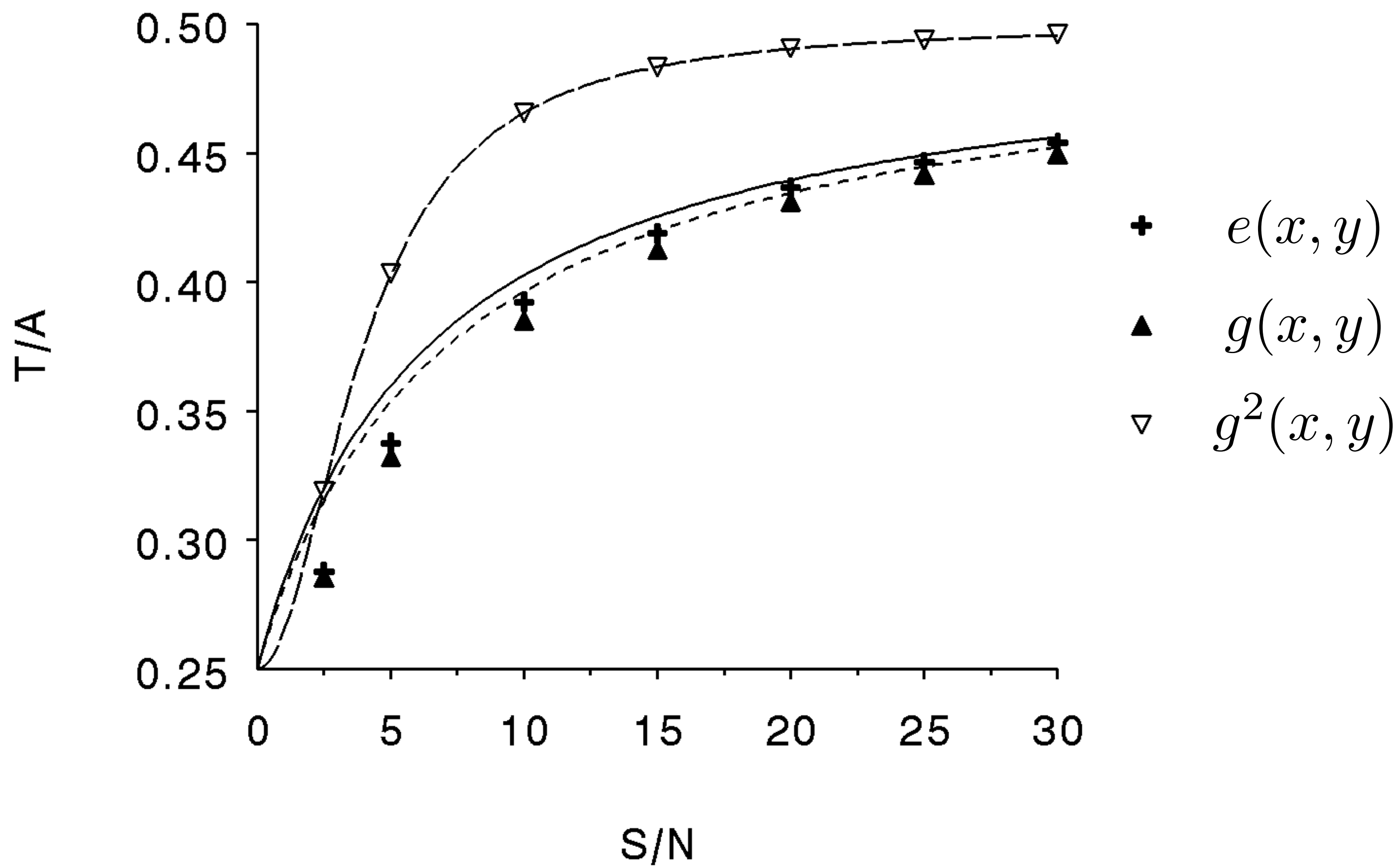
$$T_\sigma = B + \frac{A}{2} + \frac{q - \frac{1}{2}}{\frac{\epsilon g|A|}{\nu\sigma} + 1} A \tag{29}$$

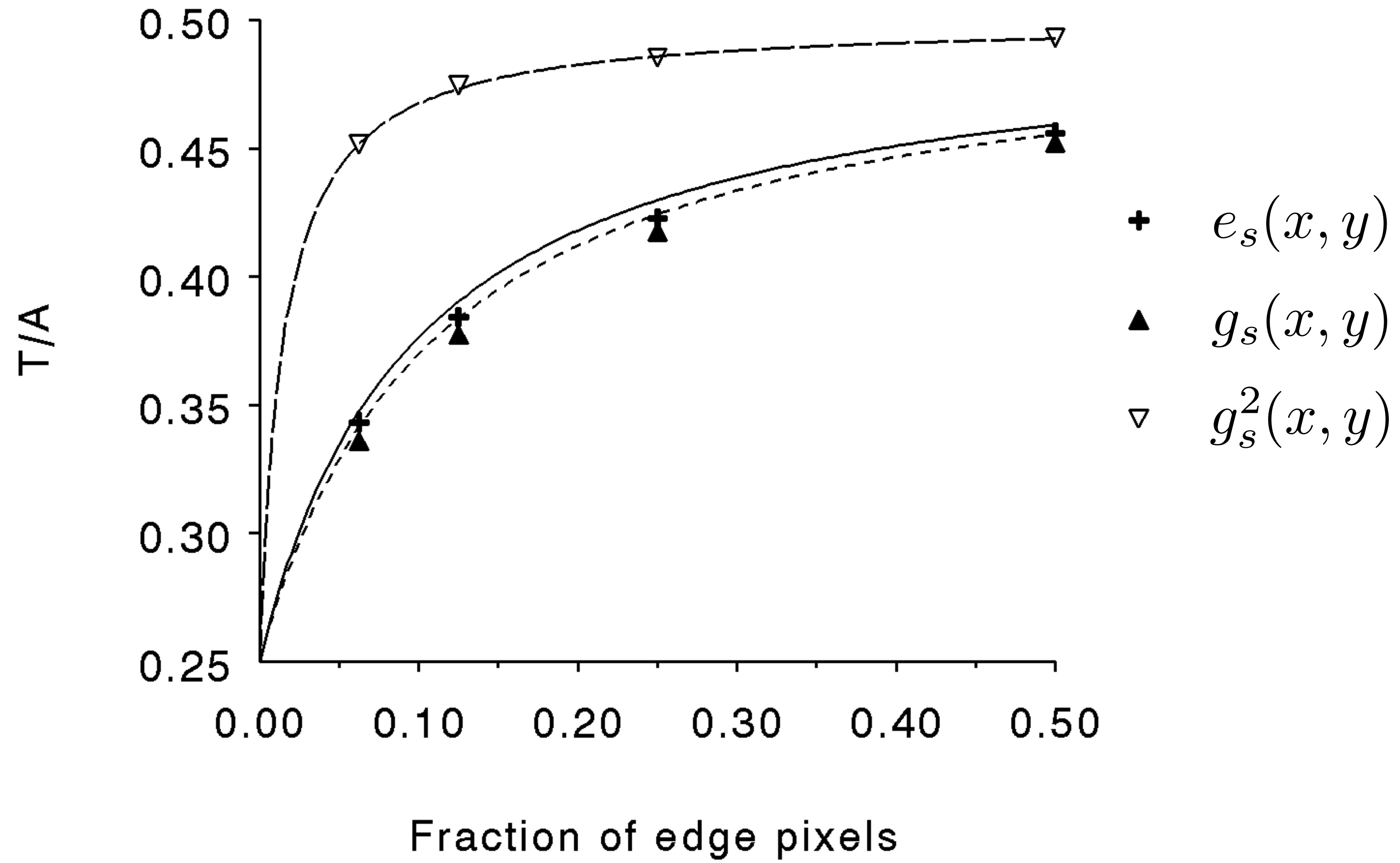
- An edge detector with high response to real edges, and low response to noise reduces the bias

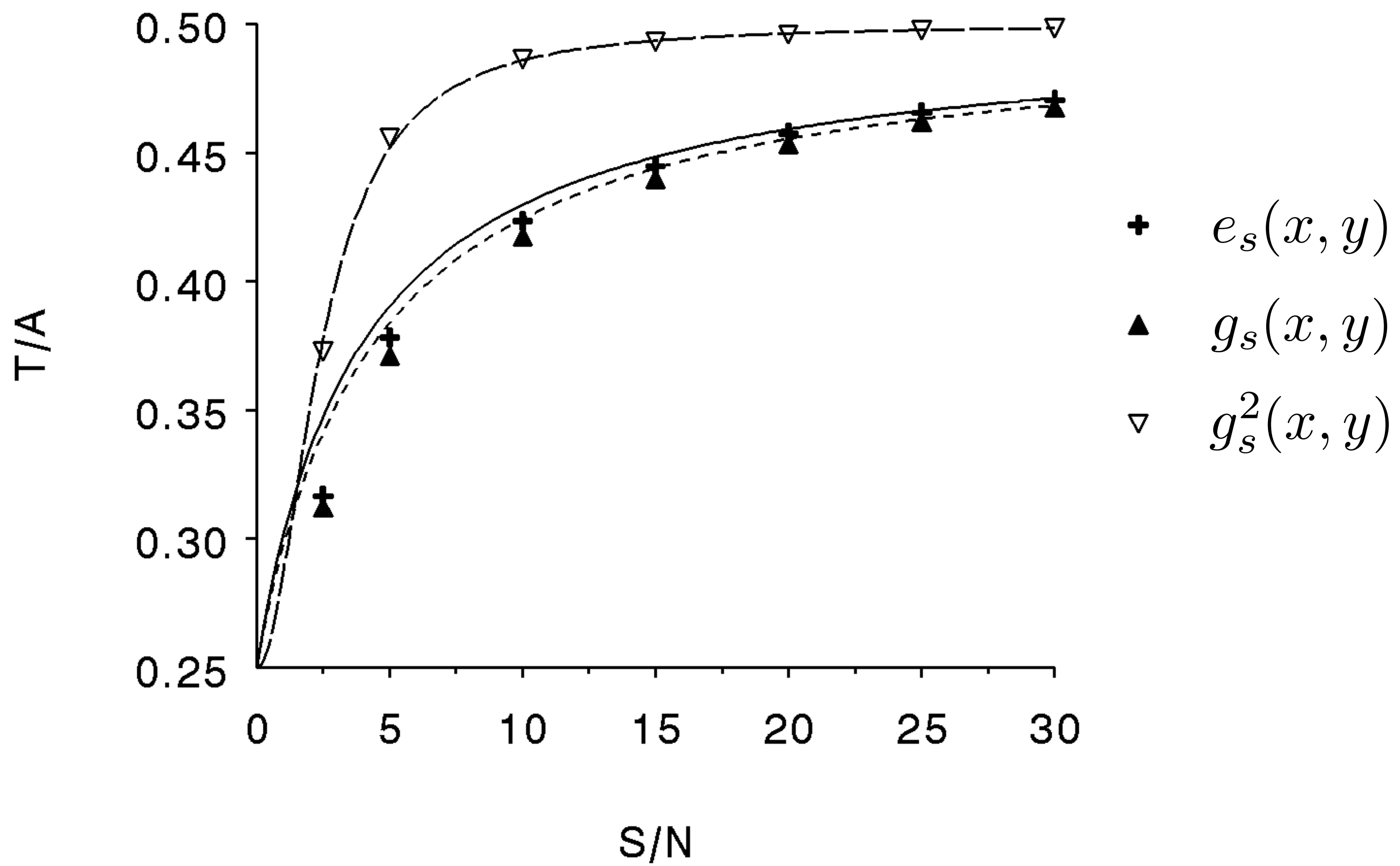
Measuring Noise Bias



Measuring Noise Bias




0.50
0.45
0.40
0.35
0.30
0.25
T/A
0.00
0.10
0.20
0.30
0.40
0.50
Fraction of edge pixels
$e_s(x,y)$
$g_s(x,y)$
$g_s^2(x,y)$


0.50
0.45
0.40
0.35
0.30
0.25
T/A
0
5
10
15
20
25
30
S/N
$e_s(x,y)$
$g_s(x,y)$
$g_s^2(x,y)$

- In the presence of noise, T is biased towards the most common category in the image.
- This noise bias is counteracted by changing T to

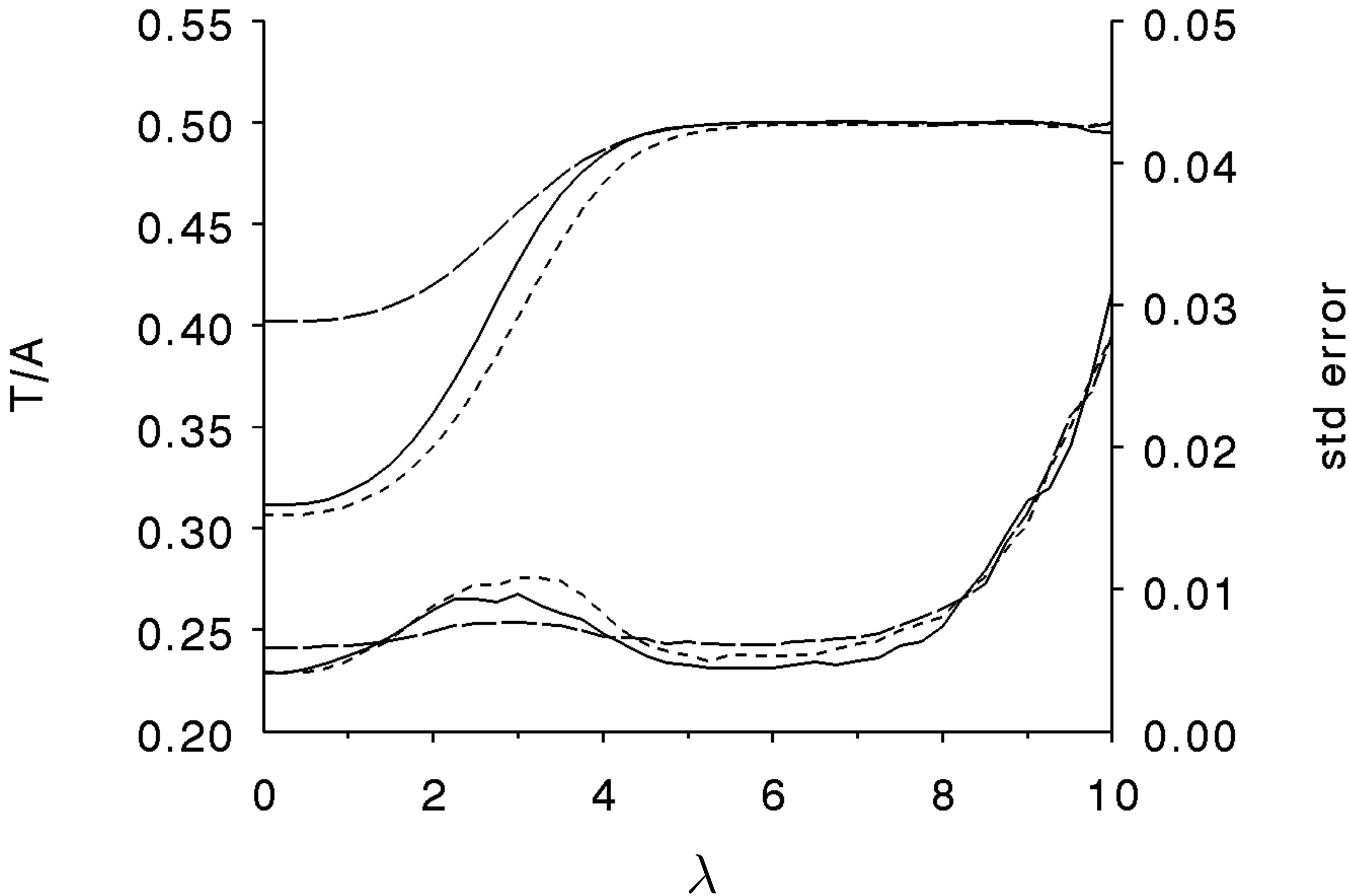
$$T = \frac{\sum w(x,y)p(x,y)}{\sum w(x,y)}, \tag{30}$$

with

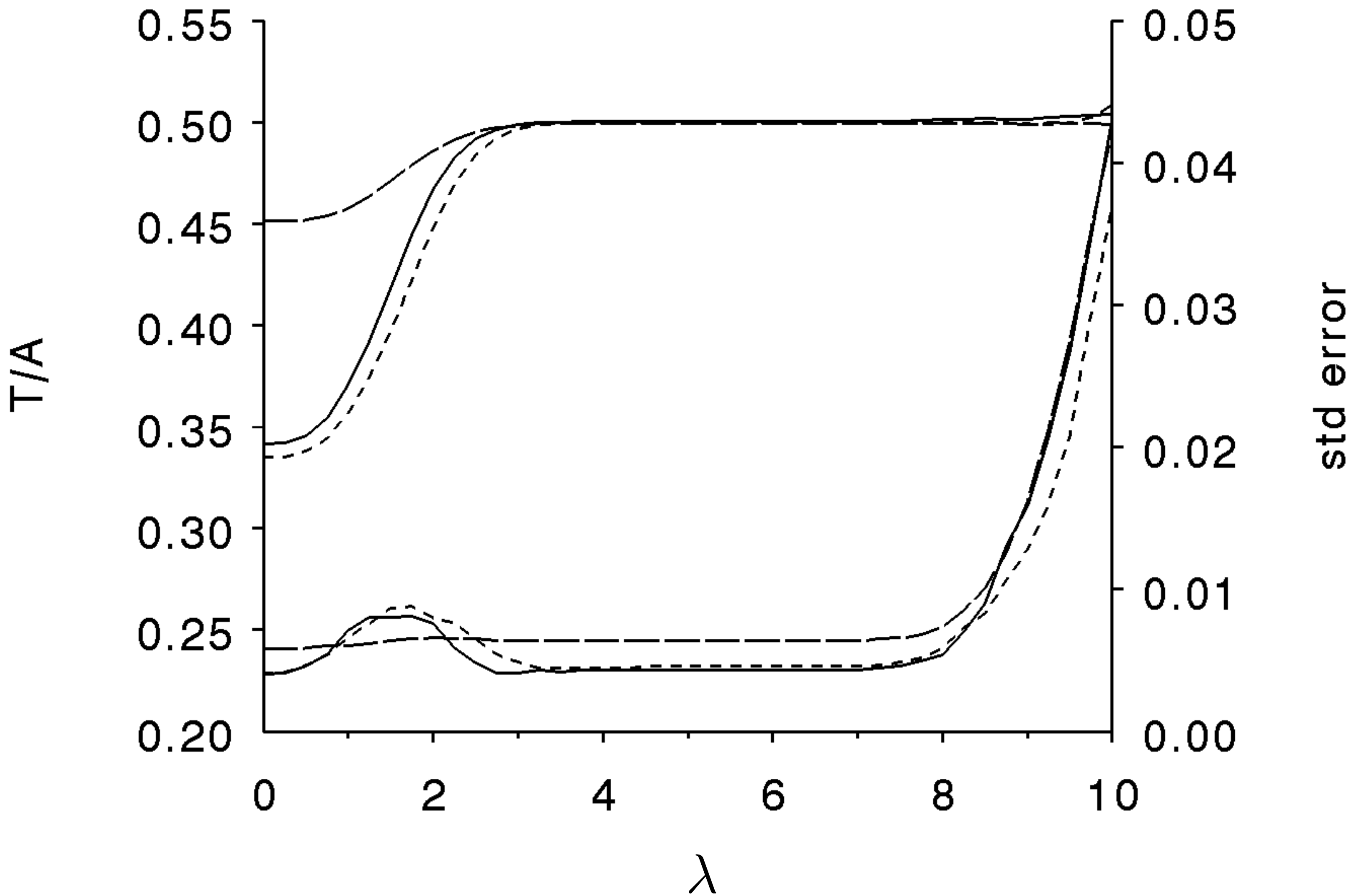
$$w(x,y) = \begin{cases} e(x,y) & \text{if } e(x,y) > \lambda\sigma \\ 0 & \text{otherwise,} \end{cases} \tag{31}$$

in which σ is the standard deviation of the image noise, and λ is an adjustable parameter.

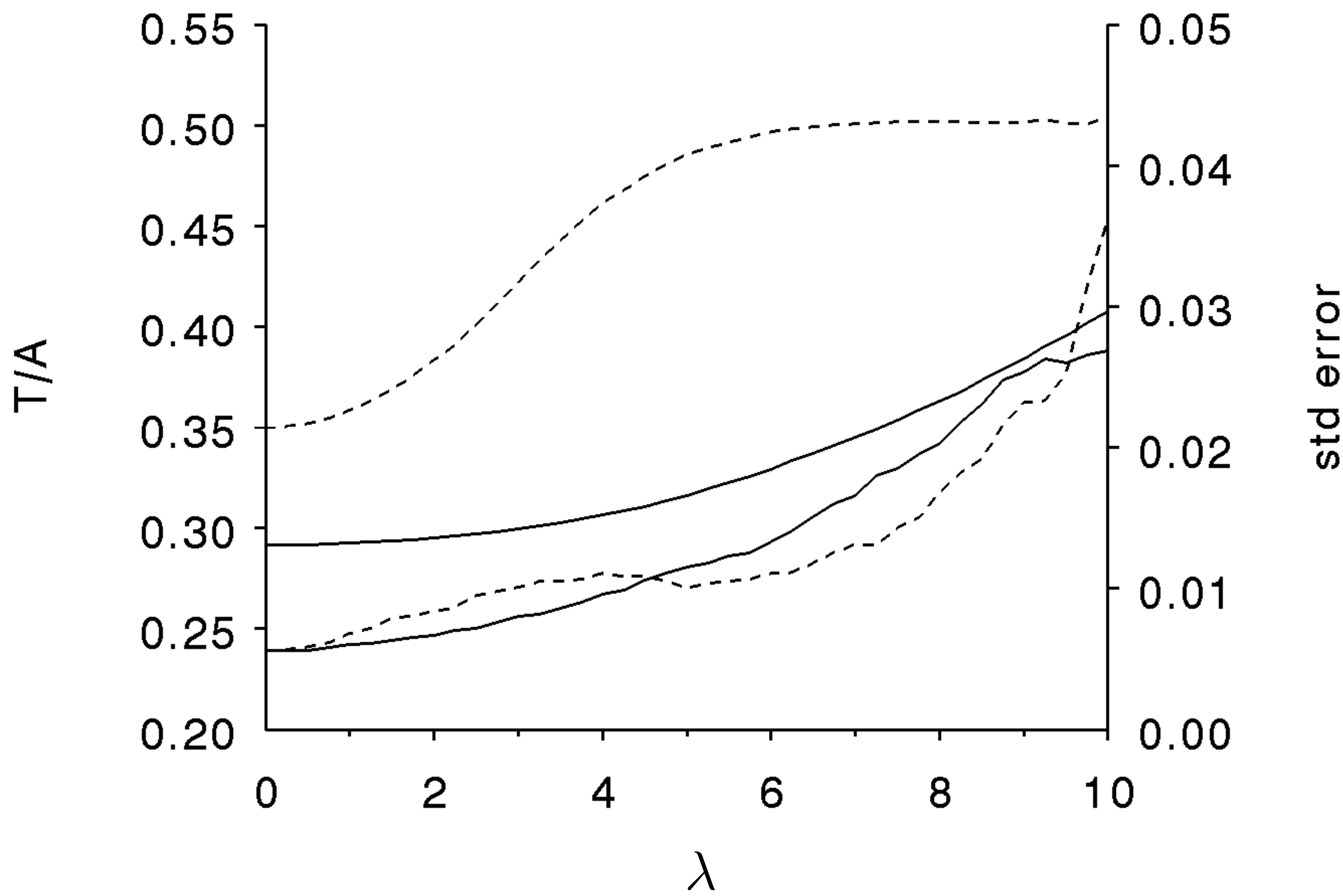
Reduction as a function of λ

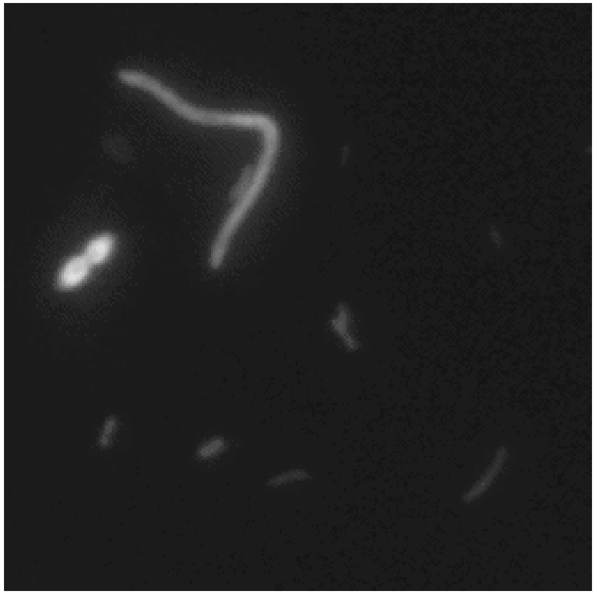


Reduction as a function of λ II

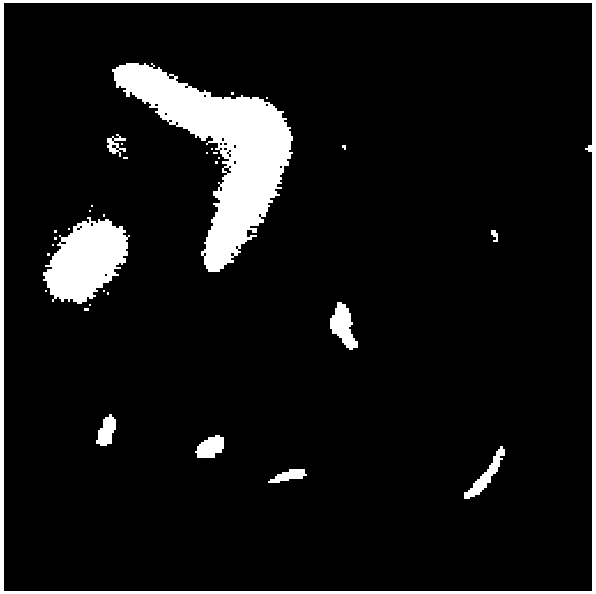


Reduction as a function of λ III





original



RATS using Eqn. 14

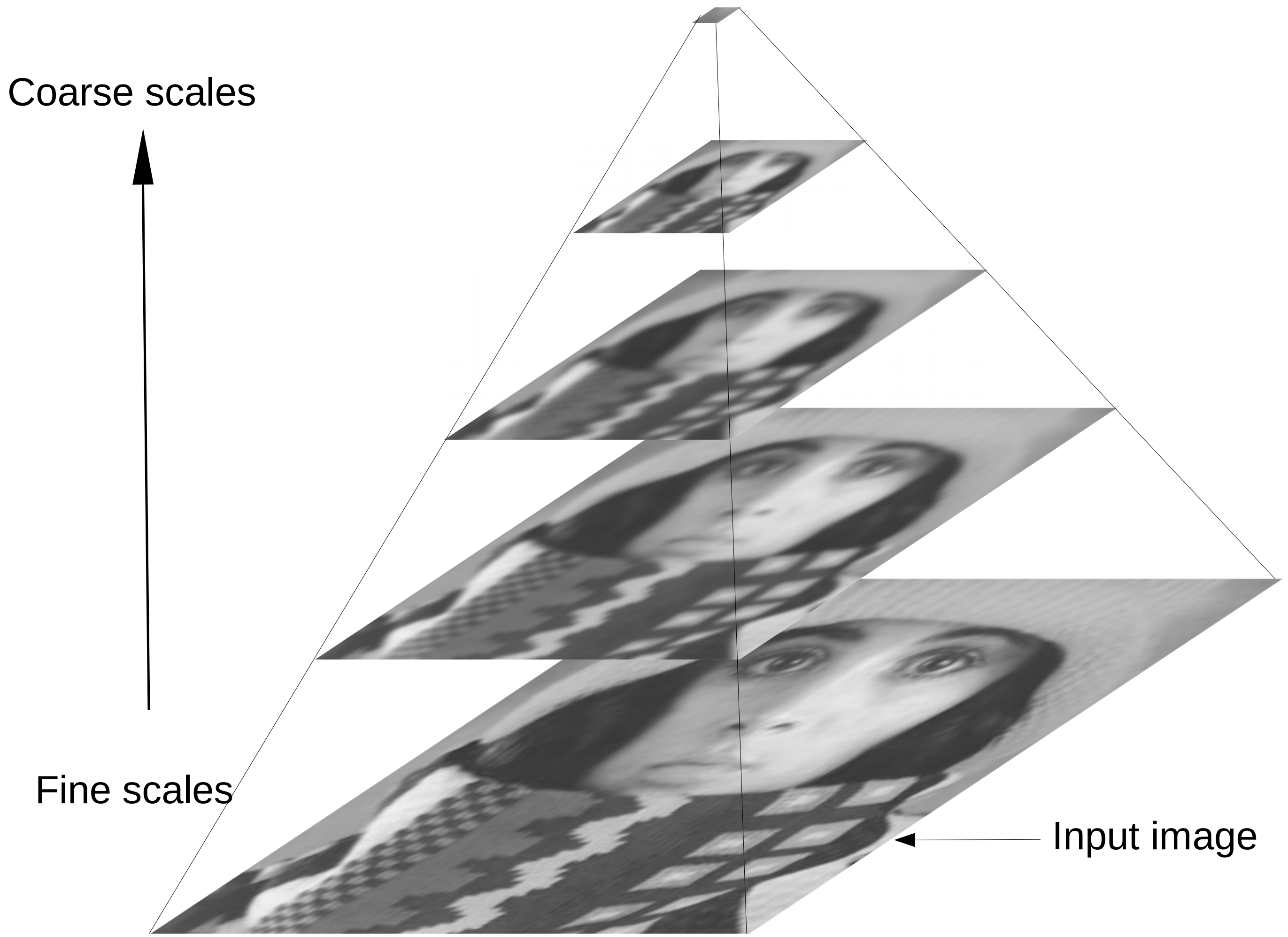
RATS using Eqn. 30

What is wrong?

- None of the thresholds so far worked well
- The problem is they are *global*
- They do not take local variations in either background or objects into account.
- Solution: compute local threshold through:

1. Quad-tree/Oct-tree methods
2. Moving windows of fixed width
3. Moving windows at multiple scales

Local Thresholding: Quad-Trees



- Compute two statistics per node n in the quad tree:
  - The threshold statistic $T(n)$
  - a measure of its reliability $R(n)$
- If reliability of root is bad, abort thresholding, otherwise
- From the root downward
  - If for the node n $R(n)$ is above some reliability threshold, keep current value of $T(n)$,
  - otherwise, assign parent threshold $T(p)$ to $T(n)$.
- Assign resulting leaf thresholds to centres of their area, and compute others by interpolation

- Otsu's method can be used with σ_B^2 or σ^2 as reliability measure
- RATS can be used with the sum of weights as reliability measure

$$R(n) = \sum_{(x,y)\in n} w(x,y) \tag{32}$$

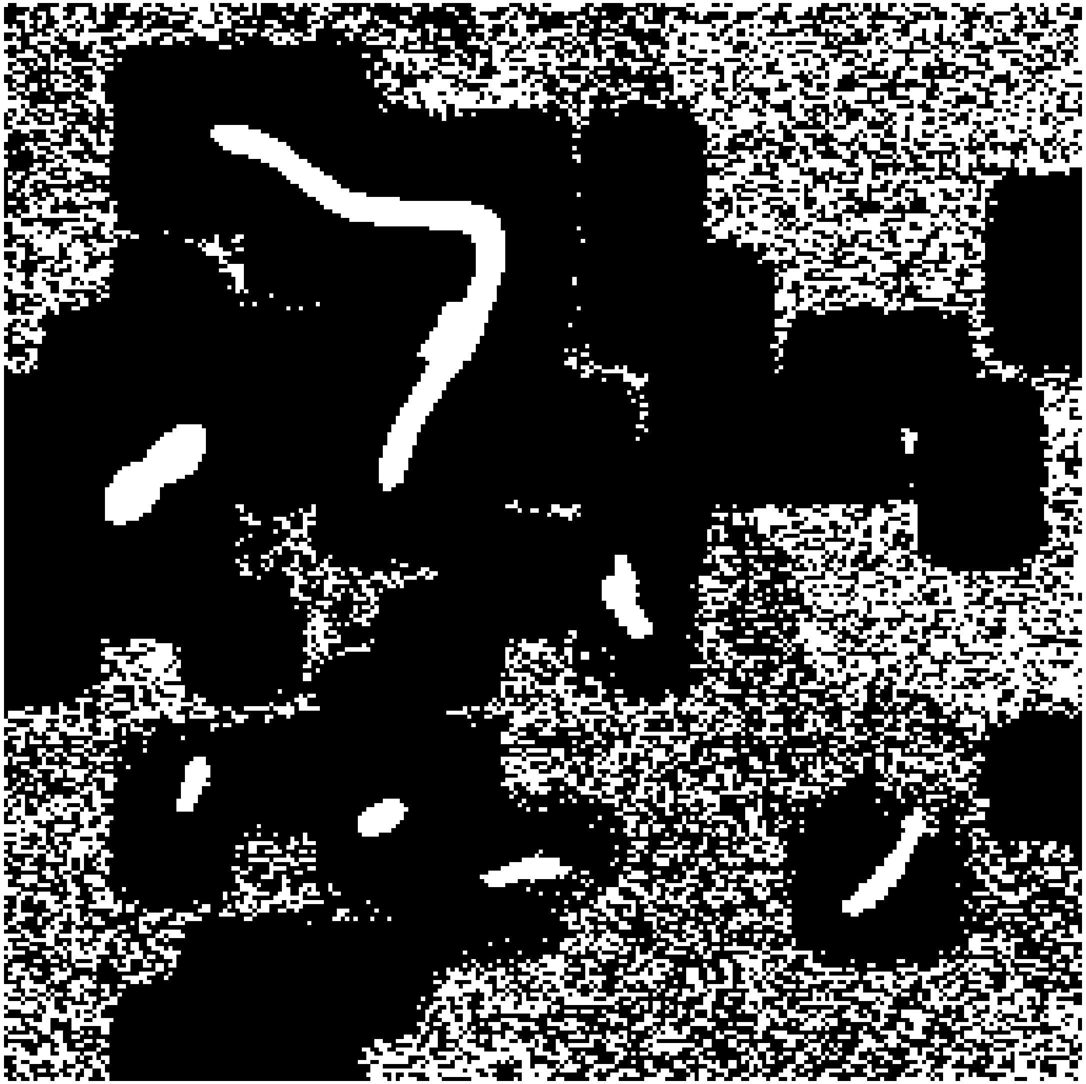
- This is particularly efficient, as we only need to compute the numerator and denominator of (30) for each leaf
- After this, each nodes statistic can be computed from the sums of the numerators of their children and the denominators of their children

Examples

quad-tree Otsu

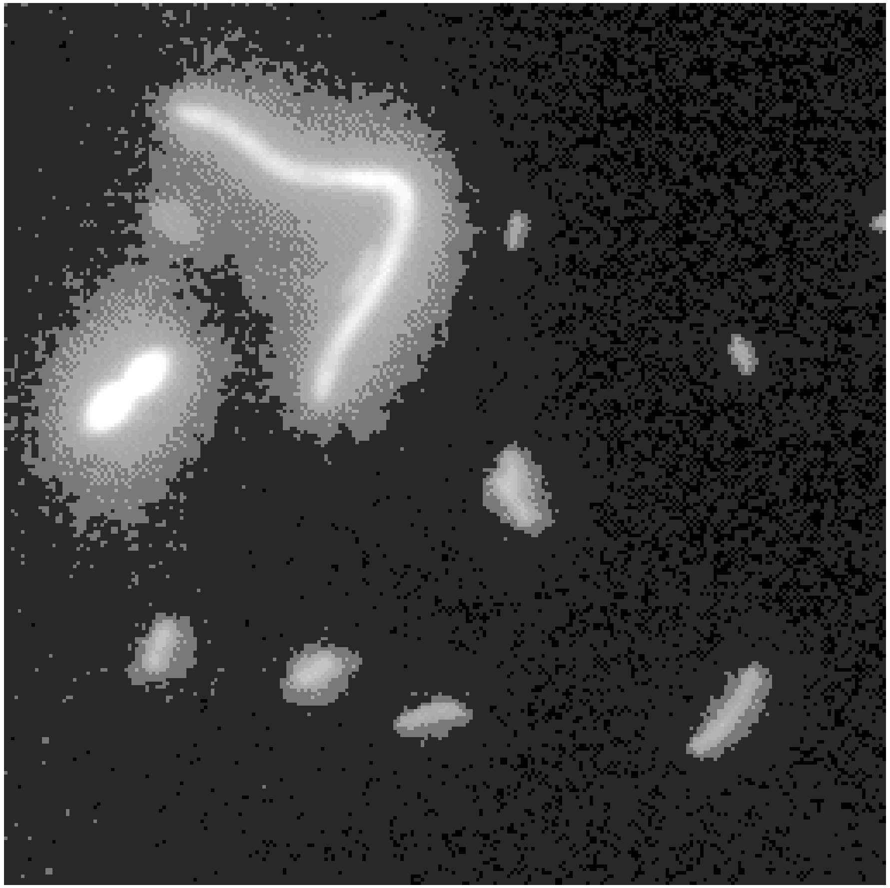
quad-tree RATS

Without reliability measure

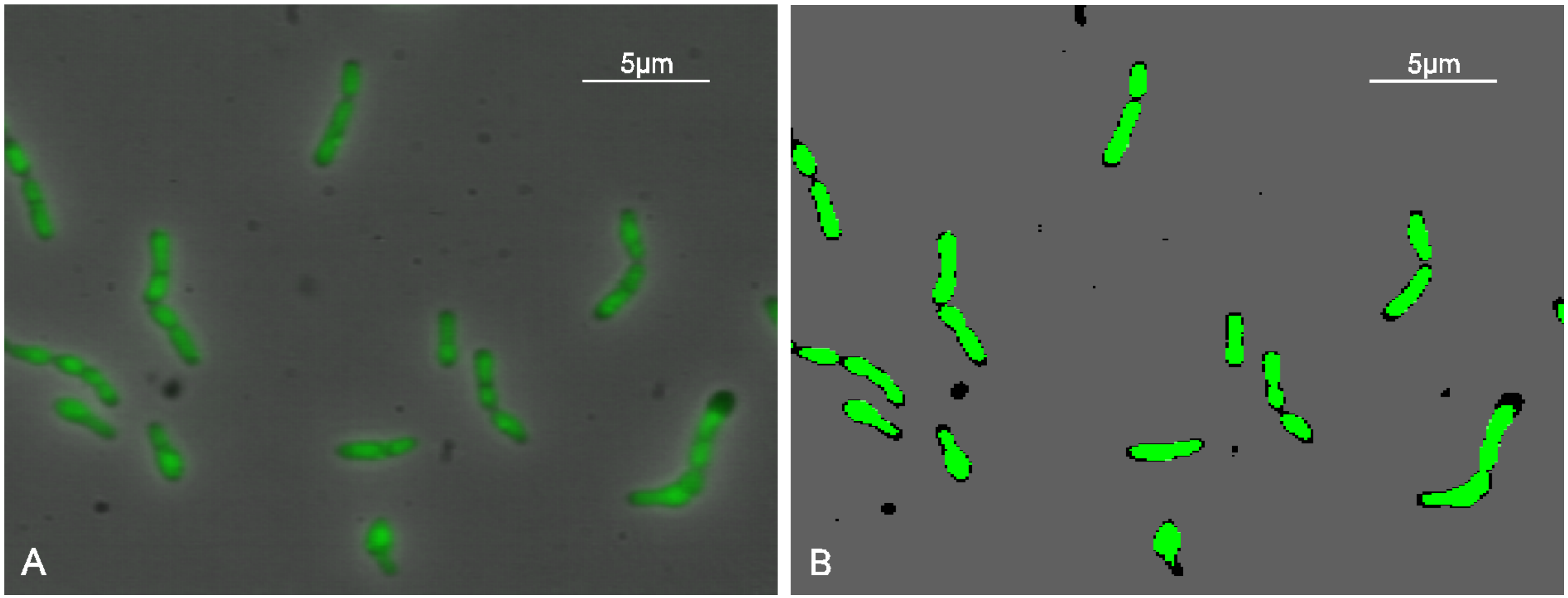


With noise removal

QT-RATS with median filter



contrast stretched original



- Quad-tree RATS has been used in bacterial image processing successfully
- It is extremely fast: 15s on a 384×512 one a 12 MHz 80286
- It works well on both phase contrast and fluorescence images

- One way to bypass the interpolation issues is by using a moving window
- The best known method for this is by Niblack
- Within a moving window $W(x, y)$ of width w centred on the current pixel (x, y) the following two statistics are computed

$$\mu(x,y) = \frac{1}{w^2} \sum_{(x',y') \in W(x,y)} f(x',y') \tag{33}$$

and

$$\sigma^2(x,y) = \frac{1}{w^2} \sum_{(x',y') \in W(x,y)} (f(x',y') - \mu(x,y))^2 \tag{34}$$

- If objects are brighter than the background the threshold becomes

$$T(x,y) = \mu(x,y) + k\sqrt{\sigma^2(x,y)} \tag{35}$$

  with k positive.

- Otherwise we use

$$T(x,y) = \mu(x,y) - k\sqrt{\sigma^2(x,y)} \tag{36}$$

- The Niblack method is quite fast, has only 2 parameters, but prior knowledge of the type of image used is needed.

The threshold surface for moving window rats can be written as

$$T(x,y) = \frac{\sum_{i=x-h}^{x+h} \sum_{j=y-h}^{y+h} w(i,j)p(i,j)}{\sum_{i=x-h}^{x+h} \sum_{j=y-h}^{y+h} w(i,j)}, \tag{37}$$

This is a the ratio of two convolutions

$$T_h(x,y) = \frac{(\Pi_h * (w \cdot p))(x,y)}{(\Pi_h * w)(x,y)}, \tag{38}$$

in which $*$ denotes convolution and $\Pi_h(x,y)$ is given by

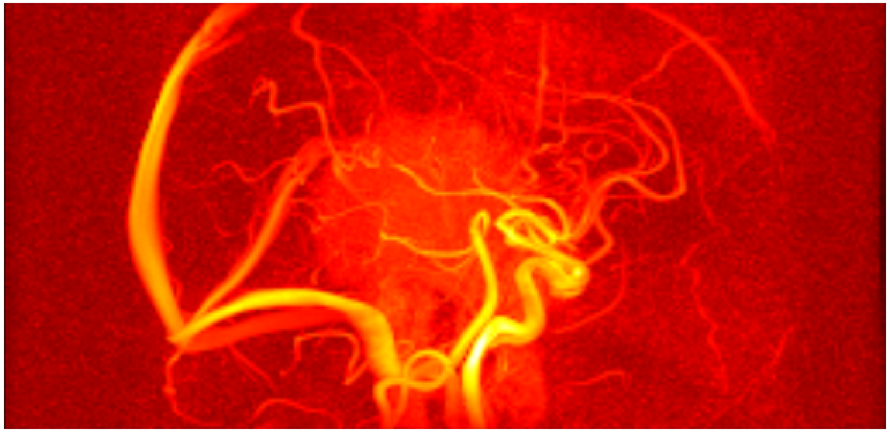
$$\Pi_h(x,y) = \begin{cases} 1 & \text{if } |x| \leq h \text{ and } |y| \leq h \\ 0 & \text{otherwise.} \end{cases} \tag{39}$$

- In (38) T is undefined for all voxels where $(\Pi_h * w)(x, y, z) = 0$.
- However, we can generalize (38) to other convolution kernels, e.g. Gaussian.
- Gaussian kernels are rotation invariant and have infinite impulse response (IIR), and so will contribute over the entire image.
- We arrive at

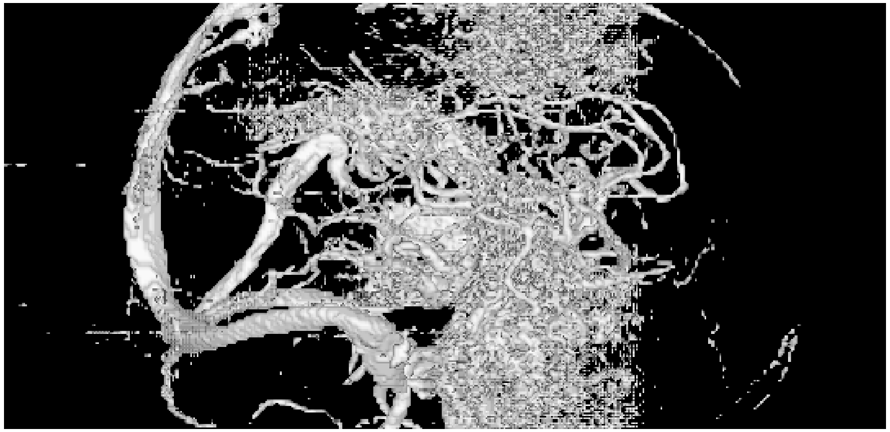
$$T_s(x, y) = \frac{(G_s * (w \cdot p))(x, y)}{(G_s * w)(x, y)}, \tag{40}$$

with G_s a Gaussian with zero mean and standard deviation of s.

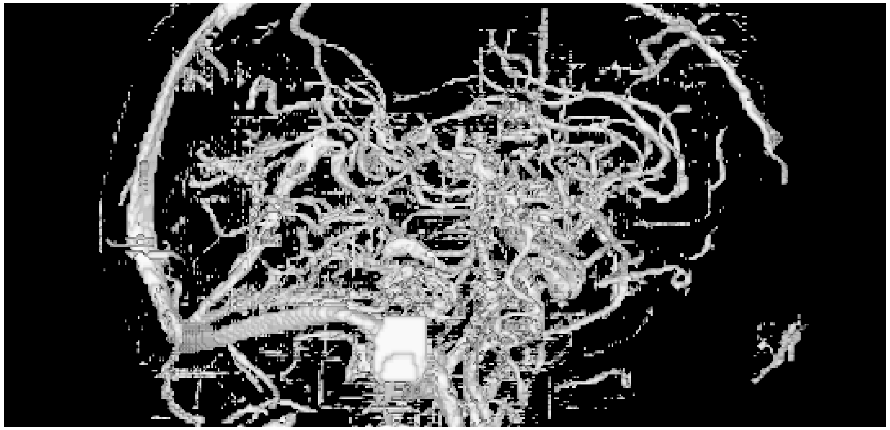
Moving Window RATS



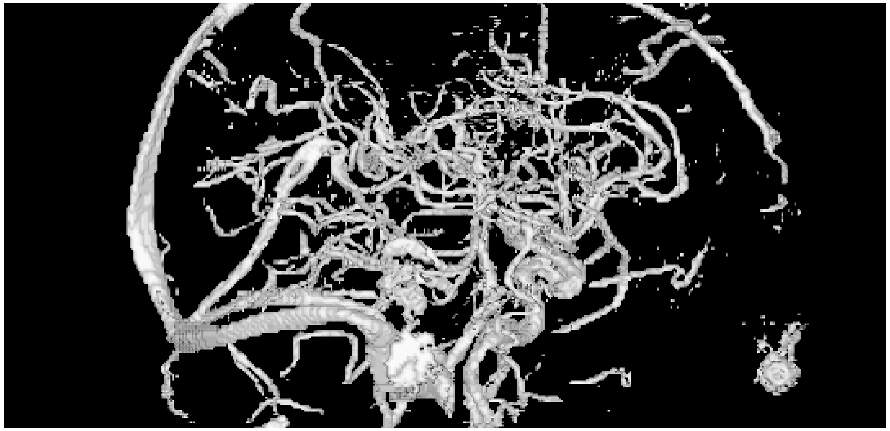
original



5-level oct-tree RATS



MW-RATS (13^3 voxel window)



GMW-RATS ($\sigma = 4$)

- Thresholding may be simple in principle, it is not trivial to get right
- Global thresholds usually fail
- Fast, local thresholders are available
- Pre- and post-processing can improve matters

?