Signal Processing

This exposition is based on:

- Steven W. Smith: *The Scientist and Engineer's Guide to Digital Signal Processing*, California Technical Publishing San Diego, California, second edition 1999. Available at http://www.dspguide.com/ [S99]

- Wikipedia articles:
  http://en.wikipedia.org/wiki/Precision_and_accuracy
  http://de.wikipedia.org/wiki/Pr%C3%A4zision (Präzision)

- Joachim Weickert: Image Processing and Computer Vision, lecture 11, 2004. [W04]

- Christian Huber: Bioanalytik, Vorlesung an der Universität des Saarlandes, 2005. [H05]

Outline

This lecture is an introduction to some of the signal processing aspects involved in the analysis of mass spectrometry data. Signal processing is a large field. Here will can only give a glimpse of it.

- Precision, accuracy, and resolution
- Morphological filters for baseline reduction
- Linear filters

Precision, accuracy, and resolution

Precision and accuracy.

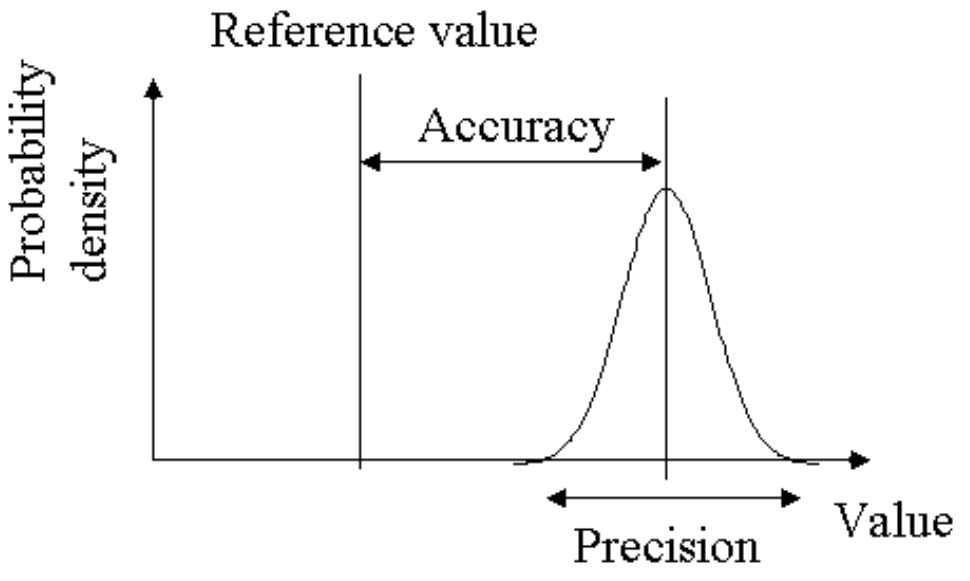
The two concepts precision and accuracy are often used interchangeably in non-technical settings, but they have very specific definitions in science engineering, engineering and statistics.

Precision, accuracy, and resolution (2)

Accuracy is the degree of conformity of a measured or calculated quantity to its actual (true) value. *Precision* is the degree to which further measurements or calculations will show the same or similar results. The results of calculations or a measurement can be accurate but not precise; precise but not accurate; neither; or both. A result is called *valid* if it is both accurate and precise.

High accuracy, but low precision

High precision, but low accuracy



Accuracy and precision

Precision, accuracy, and resolution (3)

- *Precision* is usually characterized in terms of the standard deviation of the measurements. (In German: Präzision.) Precision is sometimes stratified into:
  - *Repeatability* - the variation arising when all efforts are made to keep conditions constant by using the same instrument and operator, and repeating during a short time period.

    In German: innere Genauigkeit einer Messung, veraltet auch Wiederholgenauigkeit – die Stabilität des Messgeräts oder seiner Ablesung während des Messvorgangs selbst; dies wird durch Fehler- und Ausgleichsrechnung ermittelt nach oftmaligem Wiederholen der Messung unter gleichen Umständen und mit demselben Messgerät oder Messsystem.
  - *Reproducibility* - the variation arising using the same measurement process among different instruments and operators, and over longer time periods.

    In German: äußere Genauigkeit einer Messung – die Streuung der Messungen, wenn sie unter verschiedenen äußeren Umständen wiederholt werden.

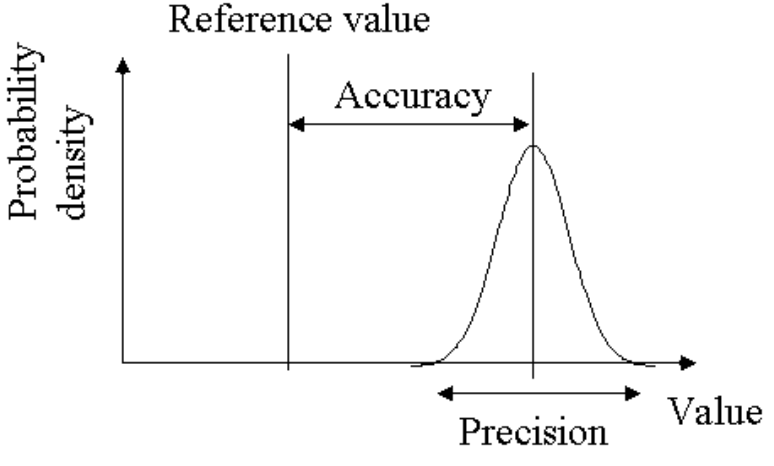
- *Accuracy* is related to the difference (*bias*) between the mean of the measurements and the reference value. Establishing and correcting for bias is the task of *calibration*, which corrects for systematic errors of the measurement. Of course this requires that a measurement of higher accuracy, or another source for the "true" value, is available!

  In German: absolute Genauigkeit einer Messung – der Grad der Übereinstimmung zwischen angezeigtem und wahrem Wert.

When deciding which name to call the problem, ask yourself two questions:

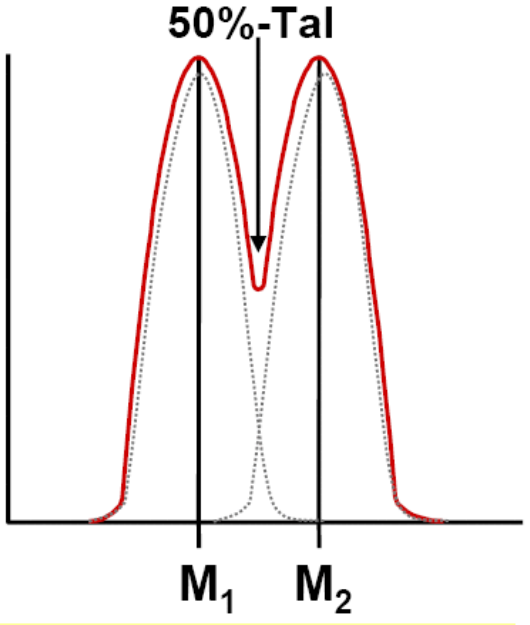
First: *Will averaging successive readings provide a better measurement?*
If yes, call the error precision; if no, call it accuracy.

Second: *Will calibration correct the error?*
If yes, call it accuracy; if no, call it precision.



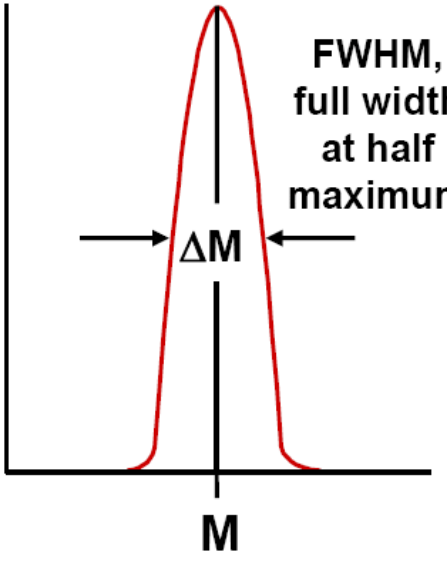
Precision, accuracy, and resolution (4)

- *Resolution* is yet another concept that is closely related to precision.



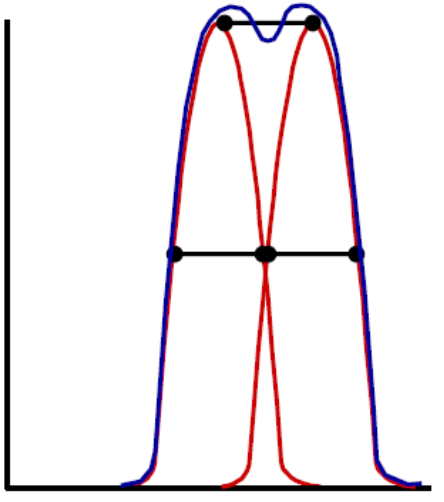
$$R_{50\%} = \frac{M_1}{M_2 - M_1} = \frac{M_1}{\Delta M}$$

$R_{50\%}$=1000 heißt, dass 1000 und 1001 mit 50% Tal aufgelöst werden



$$R_{FWHM} = \frac{M}{\Delta M}$$

R_{FWHM}=1000 heißt, dass 1000 und 1001 gerade noch aufgelöst werden, eine Auftrennung mit 50% Tal erreicht man ab etwa R_{FWHM} 1700



Precision, accuracy, and resolution (5)

The *accuracy* (in German: Massengenauigkeit) is often reported using a pseudo-unit of *ppm* (parts per million): Let

$$\Delta m\,[\mathrm{u}] = m_{\mathrm{measured}} - m_{\mathrm{theoretical}}\,.$$

Then

$$\Delta m/m\,[\mathrm{ppm}] = \frac{m_{\mathrm{measured}} - m_{\mathrm{theoretical}}}{m_{\mathrm{theoretical}}} \cdot 10^6[\,\mathrm{ppm}]\,.$$

Like accuracy, the *resolution* is a *dimensionless number*.

Morphological filters

Mathematical morphology is a relatively new branch of mathematics. It was founded around 1965 at the Ecole Normale Superieure des Mines in Fontainebleau near Paris (according to [W04]).

Morphological methods do only take into account the *level sets*

$$L_i(f) := \{ (x, y) \mid f(x, y) \geq i \}$$

of an image f.

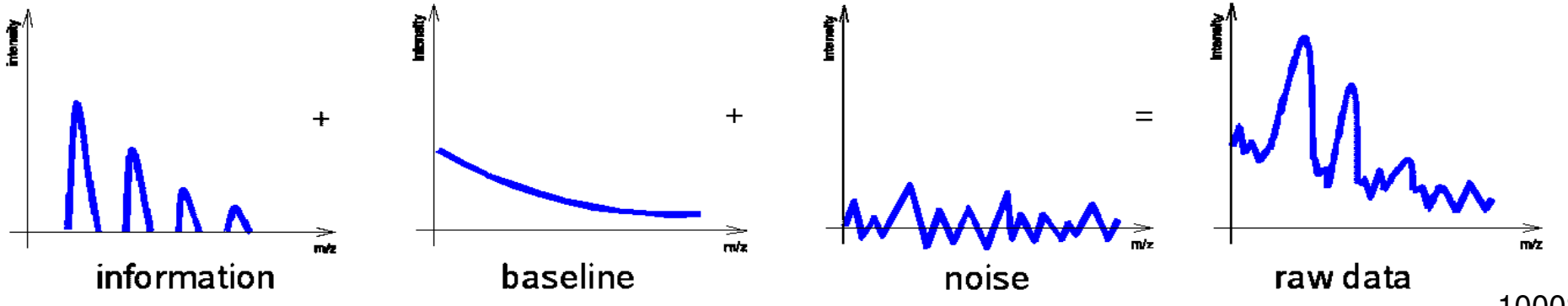
Hence the results are invariant under all strictly monotonous transformations. Morphological methods are well-suited for analyzing the *shape* of objects – this is in fact what motivates the name *morphology*. The concept of level sets is related to the *rank transformation* used in statistics.

Morphological filters provide a nonlinear alternative to the linear filters which we will learn about later.

Morphological filters (2)

Why is filtering, and signal processing in general, so important for computational mass spectrometry? A typical mass spectrum can be decomposed into three additive terms with different *'frequency range'*:

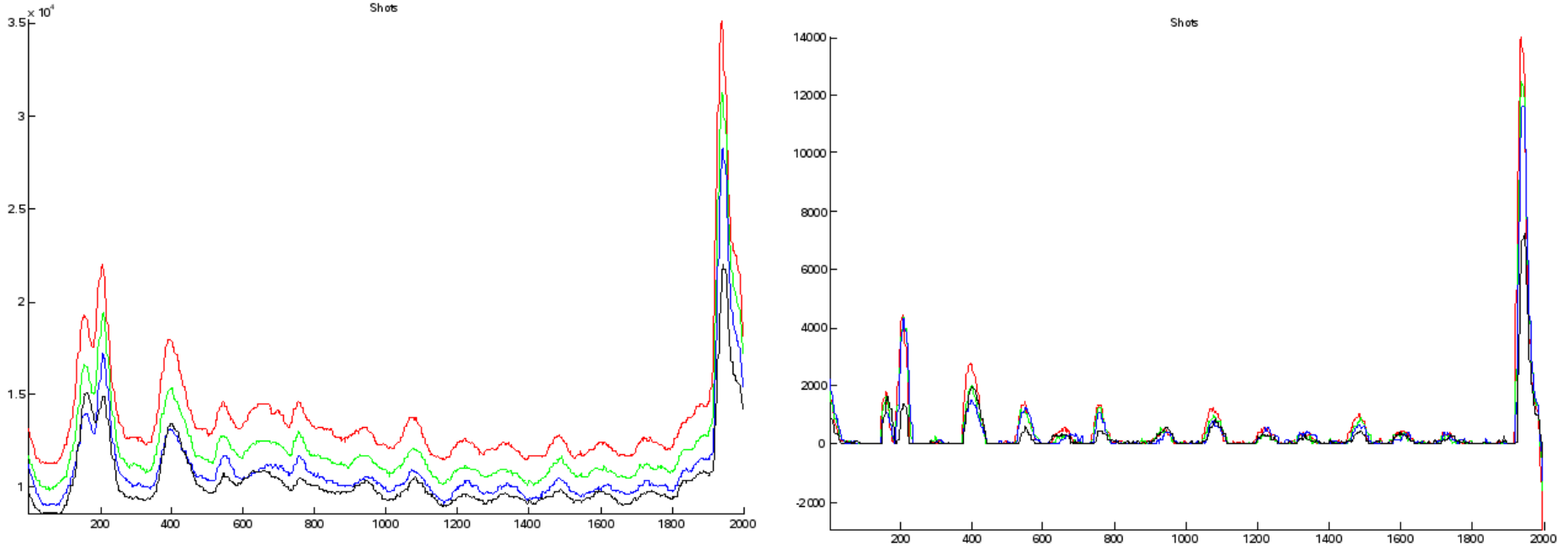
- **Information**: The "real" signal we are interested in, e.g. an isotopic pattern caused by a peptide. *Medium frequency.*
- **Baseline**: A broad trend, for example caused by signals from matrix ions when MALDI is used. *Very low frequency*, should not change much within 5 Th.
- **Noise**: *Very high frequency*, e.g. detector noise; (hopefully) not even correlated among consecutive sample points in the raw data.



Morphological filters (3)

The *top hat filter* is a morphological filter which can be used for baseline removal.

Example

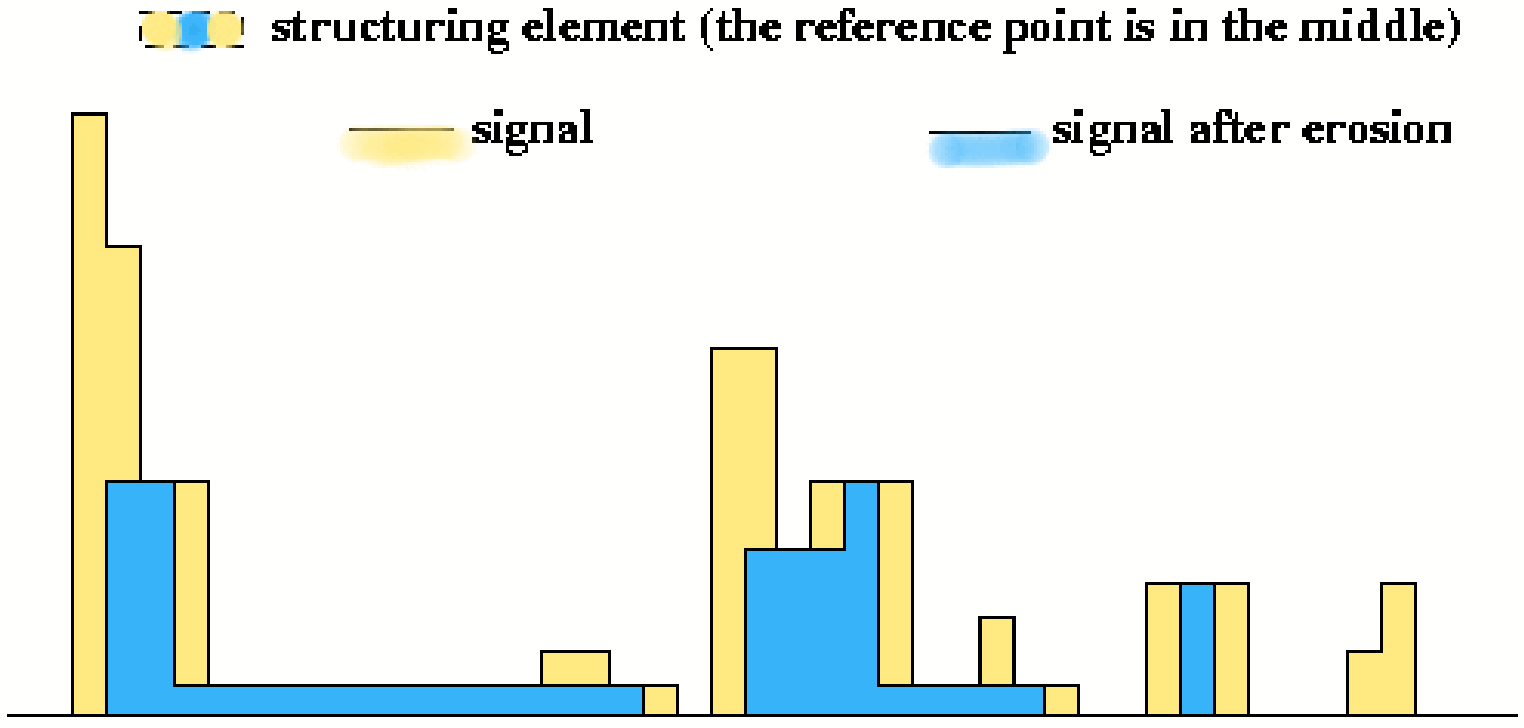


The image shows the result of a (rather aggressive) baseline reduction using the top hat filter, applied to a couple of mass spectra.

First we need to define two morphological operations: the *erosion* and the *dilation*.

Morphological filters (4)

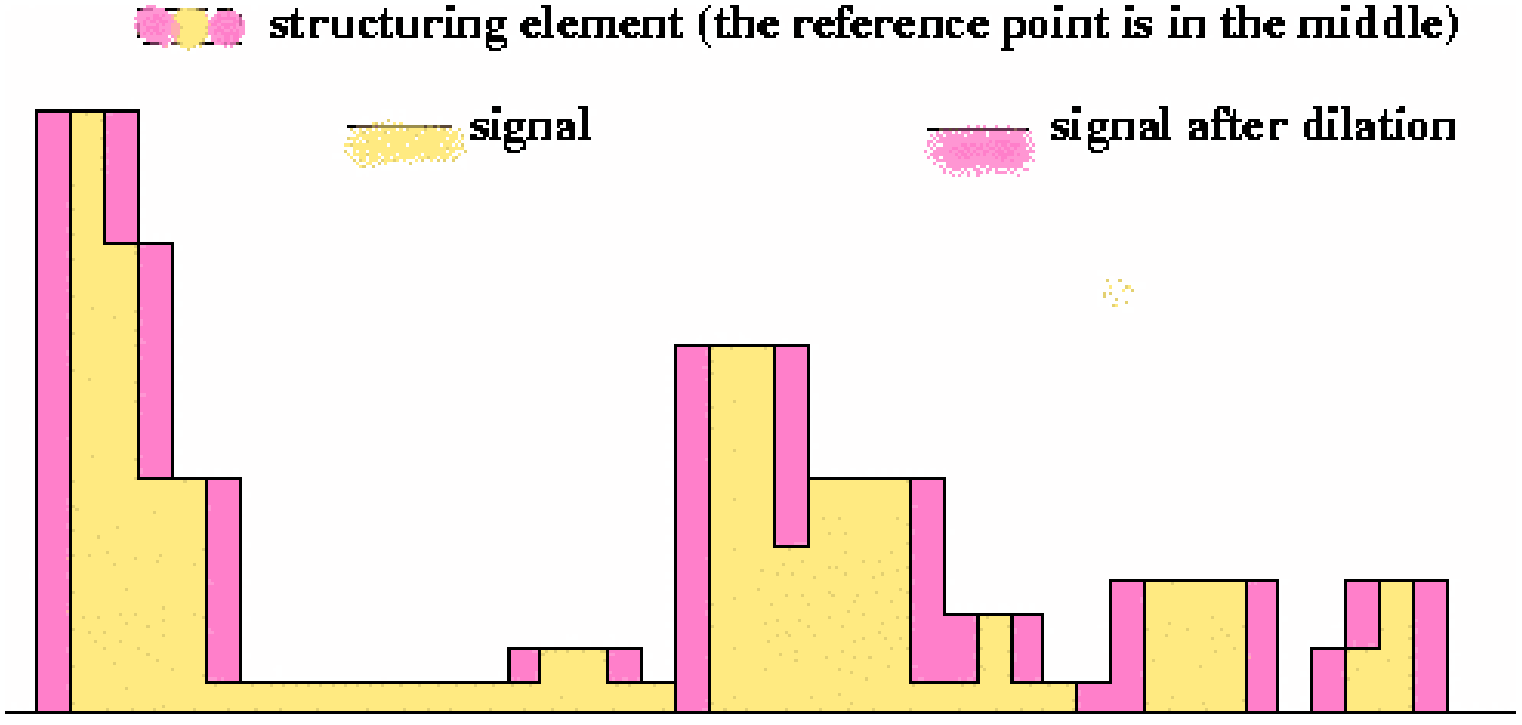
Erosion.



Intuitively, the erosion is obtained by moving the structuring element (in German: “Strukturelement”) within the area under the signal and marking the area covered by the reference point.

Morphological filters (5)

Dilation.



The dilation is defined similarly. This time we move the reference point within the area under the signal and mark the area covered by the structuring element.

In German this is called “Dila**ta**tion”, but “Dilation” also seems to be in use.

Morphological filters (6)

The mathematical definition can be given in very general terms, but in our case, for the top hat filter, we will only consider a "flat" structuring elements. Thus the structuring element B is a symmetric interval around zero, and zero is the reference point. Therefore the definitions are as simple as:

- *Dilation*: $(f \oplus B)(x) := \max\{ f(x - x') \mid x' \in B \}$.
- *Erosion*: $(f \ominus B)(x) := \min\{ f(x + x') \mid x' \in B \}$.

Since for any set X, $\min\{x \in X\} = -\max\{-x \mid x \in X\}$, it is sufficient to explain the algorithm for dilation (the max case).

Morphological filters (7)

Trivial algorithm

Assume that the signal to be processed is $x_0, x_1, \ldots, x_{n-1}$, the size of the structuring element is $p > 1$, and we want to compute the "max" filter:

$$y_i := \max_{0 \le j < p} x_{i+j} \qquad \text{for } i = 0, \ldots, n-p\,.$$

Then the trivial algorithm (using two nested "for" loops) has running time proportional to $O(np)$. If p is large, this can be impractical.

The HGW algorithm

Van Herk and (later but independently) Gil and Werman found an algorithm that requires only $3n$ comparisons, independently of the size of the structuring element. In a recent paper Gil and Kimmel improved this further to $(\frac{3}{2} + \frac{\log p}{p} + O(1/p))n$, but here will explain the simpler "$3n$ algorithm".

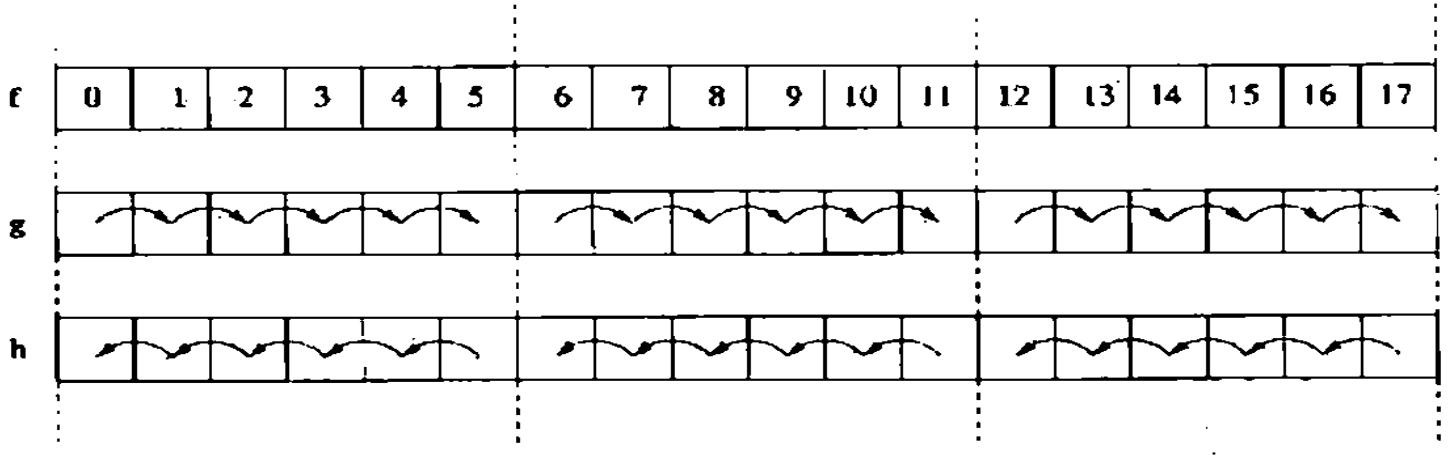
Morphological filters (8)

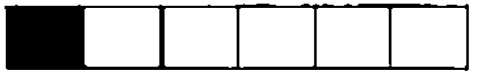
(Disclaimer: The algorithms of Van Herk and Gil-Werman have a slightly different definition of “segments”. So the following images should only be considered “exact up to ± 1”.)

Morphological filters (9)

The idea of the van Herk-Gil-Werman (HGW) algorithm is to split the input signal into overlapping *segments* of size $2p-1$, centered at $x_{p-1}, x_{2p-1}, x_{3p-1}, \ldots$.

(a) Lineares strukturierendes Element





Ergebnis(8)=max[h(8) , g(13)]

$\Rightarrow$ blackboard!!!

Morphological filters (10)

Now let j be the index of the element at the center of a certain segment. The maxima t_k of all p windows which include x_j are computed in one batch by the HGW algorithm as follows.

First, we compute *incrementally* "one half of" the maxima towards the lower indices:

$$R_k(j) := \max\{ x_j, x_{j-1}, x_{j-2}, \dots, x_{j-k} \},$$

and we do the same towards the higher indices:

$$S_k(j) := \max\{ x_j, x_{j+1}, x_{j+2}, \dots, x_{j+k} \}, .$$

Then, we *merge* the halves together and obtain the max filter by just one more maximum operation:

$$t_k := \max\{ x_{j-k}, \dots, x_j, \dots, x_{j+p-k-1} \} = \max\{R_k(j), S_{p-k-1}(j)\}$$

for $k = 1, \dots, p-2$. Moreover, we have

$$\max\{x_{j-p-1}, \dots, x_j\} = R_{p-1}(j) \qquad \text{and} \qquad \max\{x_j, \dots, x_{j+p-1}\} = S_{p-1}(j) .$$

Morphological filters (11)

The HGW algorithm operates in two stages. The auxiliary tables $R(j)$ and $S(j)$ are only needed for the current value of j, so they can be overwritten for the next segment and we will drop the index j from now on.

Preprocessing: Compute R_k and S_k from their definition. Note that $R_k = \max\{R_{k-1}, x_{j-k}\}$ and $S_k = \max\{S_{k-1}, x_{j+k}\}$ for $k = 1, \ldots, p-1$. The preprocessing requires $2(p-1)$ comparisons.

Merging: Merge the R_k and S_k together. This stage requires another $p-2$ comparisons.

The procedure computes the maximum of p windows in total. Thus the amortized number of comparisons per window is

$$\frac{2(p-1)+(p-2)}{p} = 3 - \frac{4}{p}.$$

If p is large, the preprocessing step requires about two operations per element, while the merging step requires one more such comparison.

Morphological filters (12)

The erosion and dilation operations are also defined for data in two (or even more) dimensions. The two-dimensional erosion and opening are very common operations in image processing. They have some nice mathematical properties.

- Axis-parallel *rectangular* structuring elements are *separable*, that is, erosion and dilation can be computed by applying the one-dimensional erosion algorithm to both dimensions one after another. As a consequence, the very efficient 1-D algorithms can still be applied.

- *Convex* structuring elements are *scalable*: Dilation / erosion with structuring element nB is equivalent to n dilations / erosions with structuring element B.

(Proofs: *Exercise*.)

Morphological filters (13)

Separability

We consider the case where we have two dimensions and compute the dilation. The proof for higher dimensions and/or erosion is similar.

Let f be the signal and B be the structuring element. We can assume that the reference point is 0. As we have seen, the value of the dilation of f at a certain position (x, y) is just the maximum of the original signal within the region:

$$(f \oplus B)(x, y) = \max\{f(x - x', y - y') \mid (x', y') \in B\}\,.$$

Now consider the case where B is an axis-parallel (!) rectangle $B = B_x \times B_y$. What would be the result of applying the one-dimensional dilation to both dimensions one after another?

Morphological filters (14)

From the first dilation we get

$$\max\{f(x - x', y) \mid x' \in B_x\}$$

and after the second dilation we have

$$\begin{aligned}
&\max\{\ \max\{f(x - x', y - y') \mid x' \in B_x\} \mid y' \in B_y\ \} \\
&= \max\{f(x - x', y - y') \mid x' \in B_x, y' \in B_y\} \\
&= \max\{f(x - x', y - y') \mid (x', y') \in B\} \\
&= (f \oplus B)(x, y)\,.
\end{aligned}$$

Morphological filters (15)

Scalability

Assume that B is a convex structuring element, that is:

$$p \in B \ \wedge \ q \in B \ \wedge \ \alpha \in [0,1] \ \Rightarrow \ (1-\alpha)p + \alpha q \in B\,.$$

All we need to show is that

$$p_1, \ldots, p_n \in B \ \Rightarrow \ p_1 + \ldots + p_n \in nB\,.$$

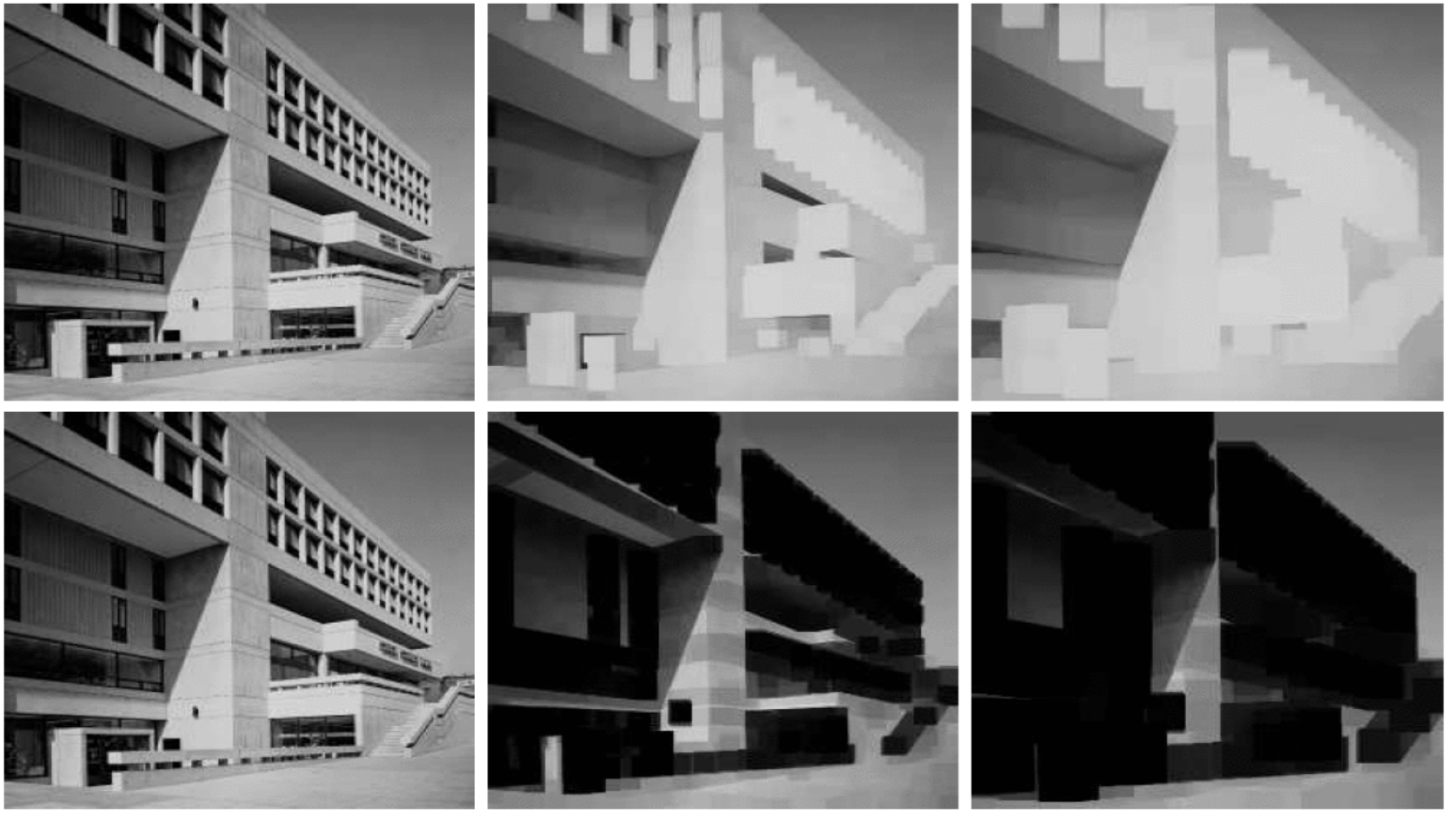
Now, observe that a simple induction using the convexity property implies that

$$p_1, \ldots, p_n \in B \ \Rightarrow \ \frac{p_1 + \ldots + p_n}{n} \in B$$

hence the assertion follows.

Morphological filters (16)

Examples from image processing



First row: Dilation of a greyscale image (256 256 pixels) with a square of length 11 and 21. – *Second row*: Same for erosion.

Morphological filters (17)

As you can see from the examples, erosion and dilation “shift” the overall intensity.

In order to simplify the image structure while avoiding the expansion effects of dilation, one can perform an erosion after the dilation. The resulting operation is called *closing*:

$$f \bullet B := (f \oplus B) \ominus B\,. \qquad \text{”closing := dilation followed by erosion”}$$

The closing removes “dark” details.

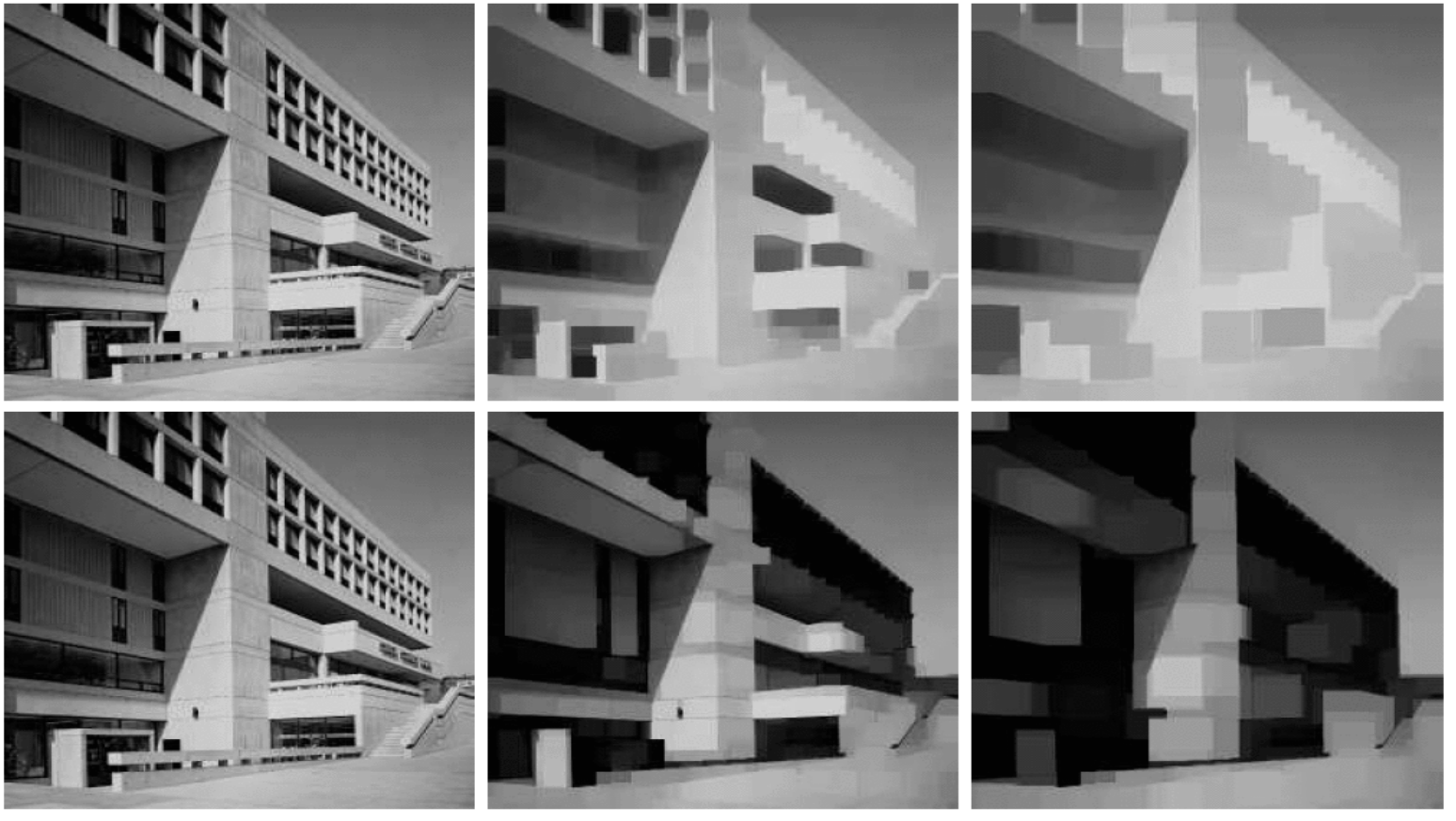
Similarly, we can simplify the image structure while avoiding the shrinkage effects of erosion by performing a dilation after the erosion. The resulting operation is called *opening*:

$$f \circ B := (f \ominus B) \oplus B\,. \qquad \text{”opening := erosion followed by dilation”}$$

The opening removes “bright” details.

Morphological filters (18)

Examples from image processing



First row: Closing of a greyscale image (256 256 pixels) with a square of length 11 and 21. – *Second row*: Same for opening.

Morphological filters (19)

More nice mathematical properties:

- The opening and closing satisfies the following inequalities:

$$f \circ B \leq f \qquad \text{"opening} \leq \text{signal"}$$
$$f \bullet B \geq f \qquad \text{"closing} \geq \text{signal"}$$

- Multiple openings or closings with the same structuring element do not alter the signal any more; we have

$$(f \circ B) \circ B = f \circ B \qquad \text{"opening twice is same as opening"}$$
$$(f \bullet B) \bullet B = f \bullet B \qquad \text{"closing twice is same as closing"}$$

(Proof: *Exercise*.)

Morphological filters (20)

To show that the opening satisfies the inequality $f \circ B = f \ominus B \oplus B \leq f$, let us assume that the contrary holds for some argument x. Let $y := f \ominus B \oplus B(x) > f(x)$. Then by definition of $\oplus$ there exists an $x' \in B$ such that $f \ominus B(x - x') = y$. But the definition of $\ominus$ implies that for all $x'' \in B$ it holds $f((x - x') + x'') \geq f \ominus B(x - x')$. If we set $x'' = x'$, then $f(x) \geq f \ominus B(x - x') = y > g(x)$, a contradiction. – In the same way, one can prove that the closing satisfies the inequality $f \bullet B \geq f$.

To show that the opening satisfies $f \circ B \circ B = f \circ B$, we prove that in fact a slightly stronger assertion holds: If $g = h \oplus B$ is the result of a dilation by B, then $g \circ B = g$. (In our case, $h = f \ominus B$.) By the preceding inequality, it suffices to show that $g \circ B \geq g$. Let x be arbitrary, then the claim is that $g \ominus B \oplus B(x) \geq g(x)$. By the definition of $\oplus$, there is an $x' \in B$ such that $g(x) = h(x - x')$. But then by the definition of $\oplus$ we also have $h \oplus B((x - x') + x'') \geq h(x - x')$ for all $x'' \in B$. Hence by the definition of $\ominus$, we have $h \oplus B \ominus B(x - x') \geq h(x - x')$. Again by the definition of $\oplus$, we have $h \oplus B \ominus B \oplus B((x - x') + x''') \geq h(x - x') = g(x)$ for all $x''' \in B$. Now let $x''' = x'$. It follows that $h \oplus B \ominus B \oplus B(x) \geq g(x)$, as claimed. – In the same way, one can prove that the closing satisfies $f \bullet B \bullet B = f \bullet B$.

Morphological filters (21)

Opening and closing act as morphological *lowpass* filters. They remove the small details.

We obtain a morphological *highpass* filter by subtracting the lowpass filtered signal from the original signal. This is called the *top hat* filter. There are two versions:

- "White" top hat, *top hat by opening*:

$$\mathrm{WTH}(f) := f - (f \circ B)\,.$$

  It extracts small "bright" structures.

- "Black" top hat, *top hat by closing*, also known as *bot hat*:

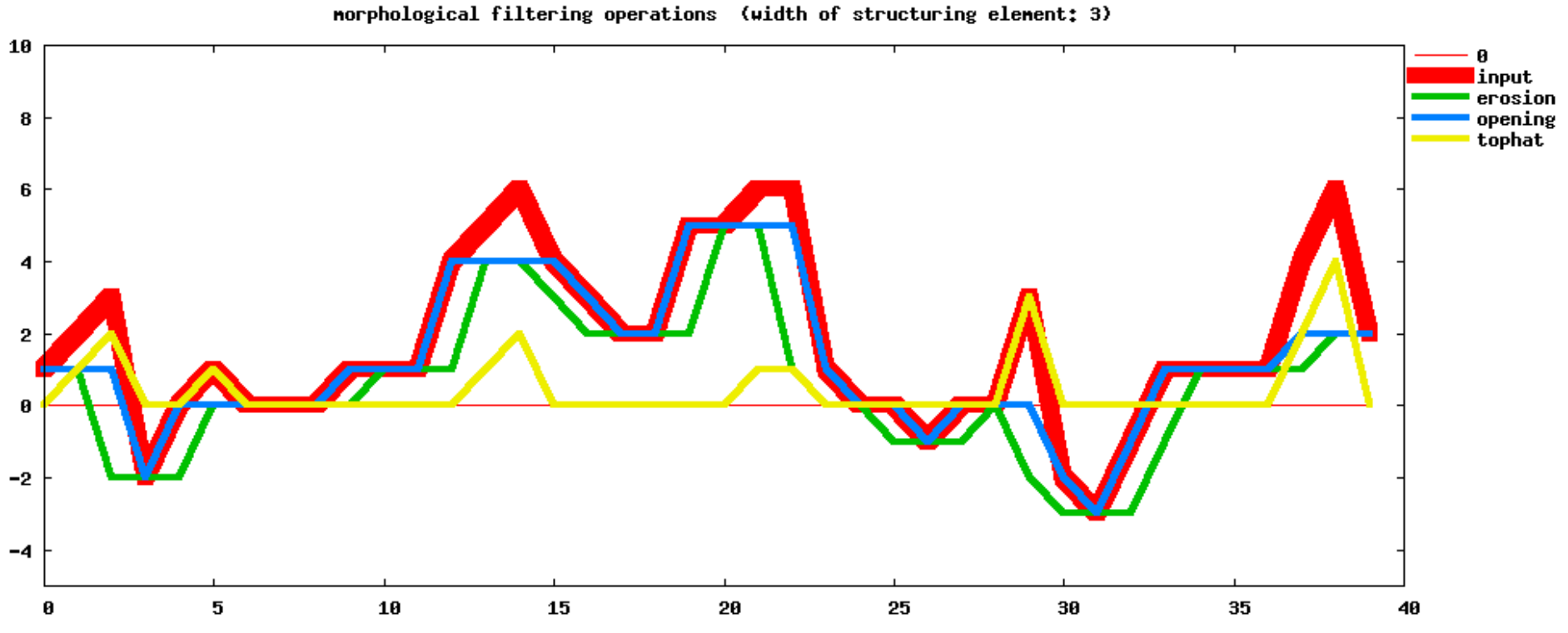
$$\mathrm{BTH}(f) := f - (f \bullet B)\,.$$

  It extracts small "dark" structures.

For mass spectra, the "white" top hat (top hat by opening) is useful to remove the baseline from the raw data.

Morphological filters (22)

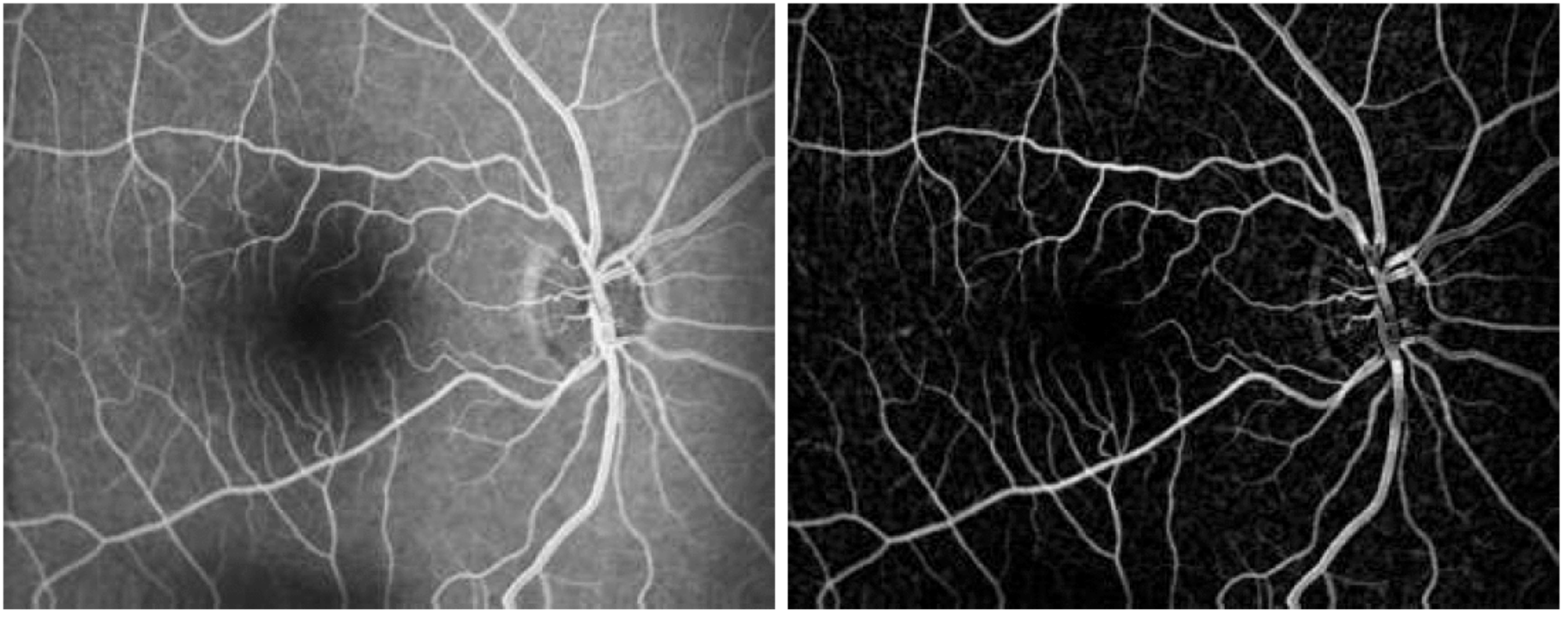
input: red, erosion: green, opening: blue, tophat: yellow



(From the documentation of class OpenMS::MorphologicalFilter.) http://www.openms.de

Morphological filters (23)

Example from image processing

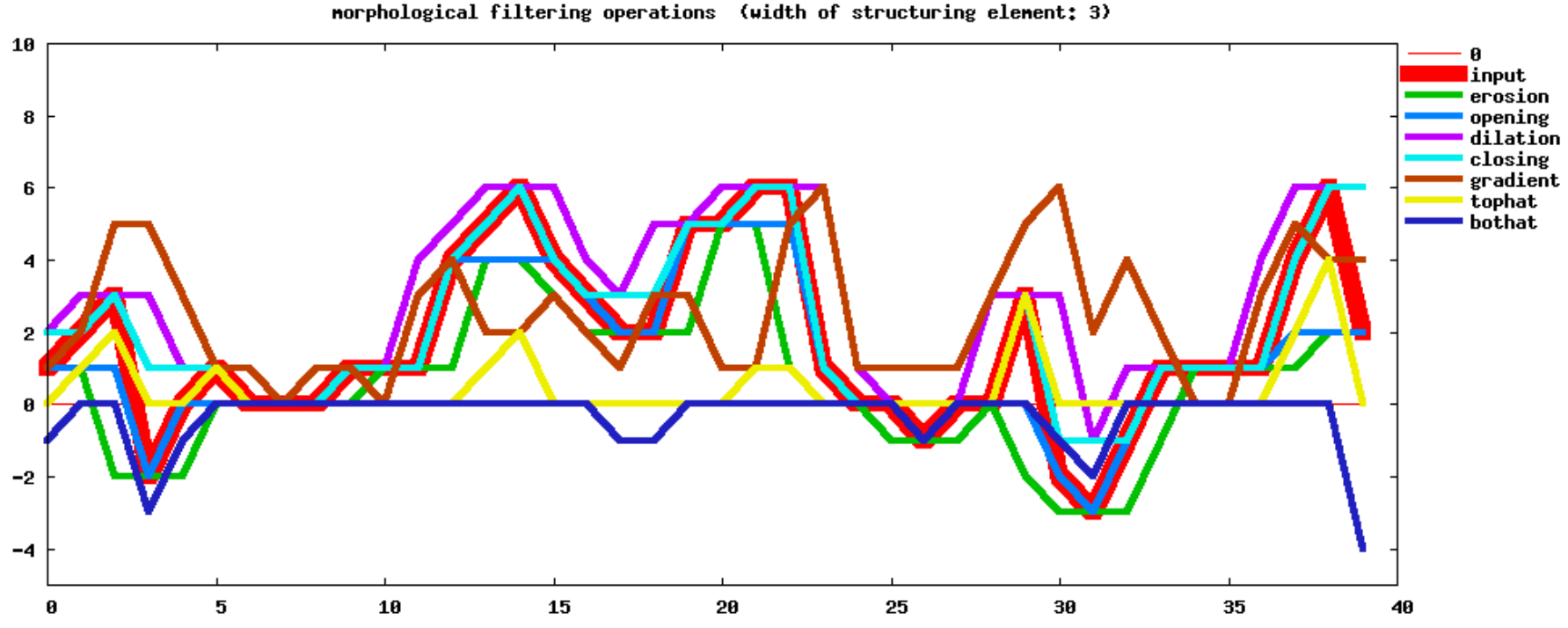


(a) **Left:** Image of the background of an eye. (b) **Right:** Vessel extraction with a white top hat.

Morphological filters (24)

What else can be done?

- A few other morphological operations . . .



- Using structuring elements of increasing size, one can remove small-, middle-, and coarse-scale structures step by step. This is called a *granulometry*. Granulometries act as morphological *bandpass* filters.

Signal processing basics

- A *signal* is a description of how one parameter varies with another parameter.

  Signals can be *continuous* or *discrete*.

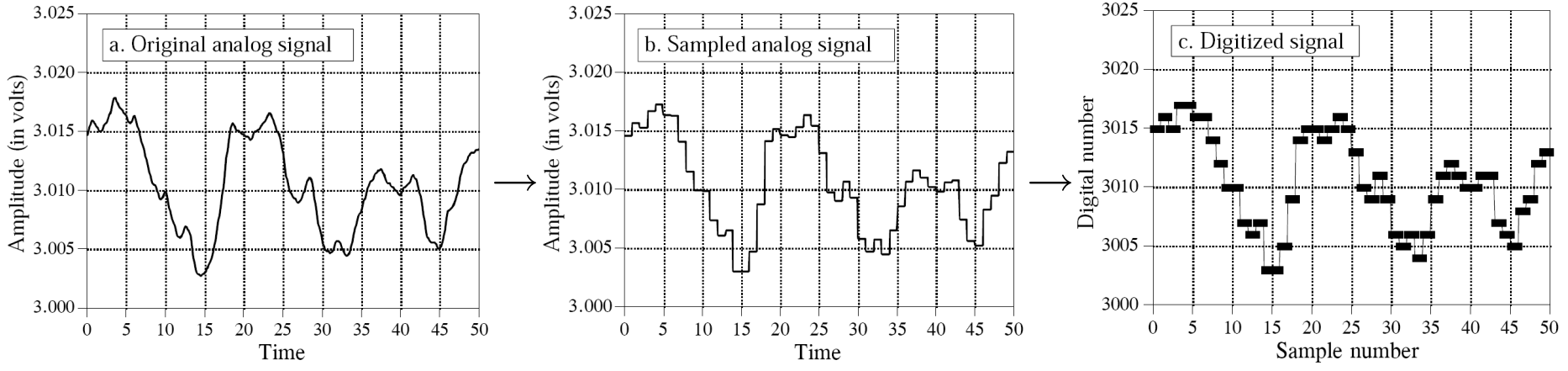
- A *system* is any process that takes an input signal and produces an output signal.

  Depending on the type of input and output, one can distinguish continuous systems, such as analog electronics, and discrete systems, such as computer programs that manipulate the values stored in arrays.

- Converting a continuous signal into a discrete signal introduces *sampling* and *quantization* errors, due to the *discretization* of the abscissa (x-axis) and ordinate (y-axis).

Signal processing basics (2)

Sampling and Quantization.



For example, a time-of-flight (TOF) mass spectrometer has to wait until the ions arrive at the detector after they have been generated and accelerated.

In a quadrupole mass analyzer, the frequency of the oscillating electrical fields determines the filtered mass-to-charge ratio of ions which arrive at the detector. The mass spectrum is obtained by scanning through the mass range of interest, which takes some time.

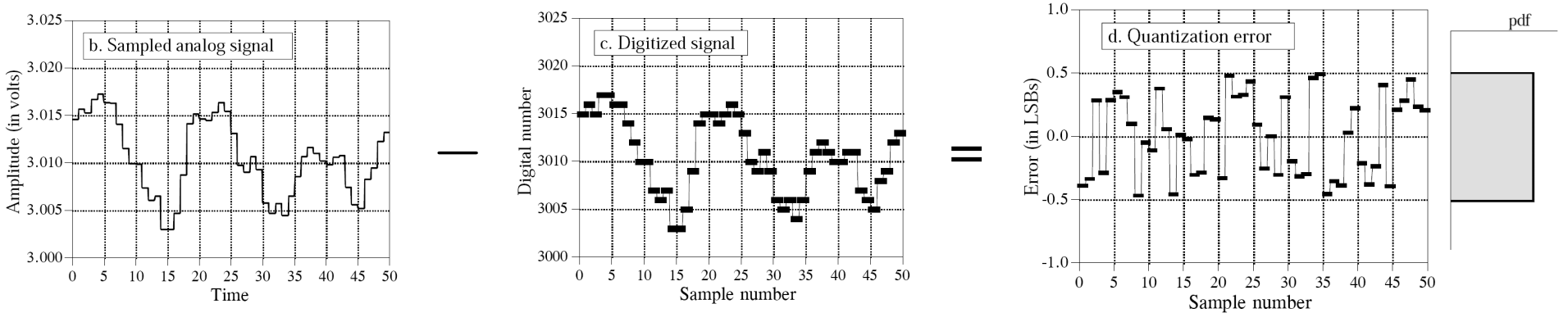
Signal processing basics (3)

From a theoretical point of view, sampling is necessary since we cannot store (or process in digital form) an infinite amount of data.

The intensity values are often reported as ‘ion counts’ but this does not mean that we do not have a quantization step here! Firstly, we should keep in mind that these ‘counts’ are the result of internal calculations performed within the instrument (e.g., the accumulation of micro-scans). Secondly, a mass spectrometer is simply not designed to count single ions but rather to measure the concentration of substrates in a probe.

Signal processing basics (4)

Fortunately, the *quantization error* is well-behaved: The digital output is equivalent to the continuous input, plus a quantization error.



In any reasonable case, the quantization error is uniformly distributed over the interval $[-\frac{1}{2}\,\mathrm{LSB}\,, +\frac{1}{2}\,\mathrm{LSB}\,]$. The term *"least significant bit"* (LSB) is a jargon for the distance between adjacent quantization levels. Thus we can easily calculate the standard deviation of the quantization error: It is $\sqrt{1/12}\cdot\ \mathrm{LSB}\ \doteq 0.288675\cdot\ \mathrm{LSB}$. (Proof: *Exercise*.)

Signal processing basics (5)

Let $X(i)$ denote the quantization error of the i-th sample. We consider X as a random variable defined on the sample numbers. It is uniformly distributed on the interval $[-\frac{1}{2}\delta, +\frac{1}{2}\delta]$, where $\delta :=$ LSB. The probability density function of X is

$$f(x) = \begin{cases} \frac{1}{\delta} & \text{for } |x| \leq \frac{1}{2}\delta \\ 0 & \text{otherwise} \end{cases}$$

By elementary calculus, we have

$$E(X) = \int_{-\frac{1}{2}\delta}^{\frac{1}{2}\delta} \frac{x}{\delta}\, dx = \frac{1}{\delta} \cdot \frac{1}{2}x^2 \Big|_{x=\frac{-\delta}{2}}^{x=\frac{\delta}{2}} = \frac{1}{\delta} \cdot \frac{1}{2}\left((\frac{\delta}{2})^2 - (\frac{-\delta}{2})^2 \right) = 0$$

Thus $V(X) = E(X^2) - E(X)^2 = E(X^2)$. Again by elementary calculus, we have

$$E(X^2) = \int_{-\frac{1}{2}\delta}^{\frac{1}{2}\delta} \frac{x^2}{\delta}\, dx = \frac{1}{3\delta}x^3 \Big|_{x=\frac{-\delta}{2}}^{x=\frac{\delta}{2}} = \frac{1}{3\delta}\left(\frac{\delta^3}{8} - \frac{(-\delta)^3}{8} \right) = \frac{\delta^2}{12}$$

Hence the standard deviation of the quantization error X is $\delta/\sqrt{12} =$ LSB $/\sqrt{12}$.

Linear systems

A **linear systems** satisfies the following properties:

- *Homogeneity*: If an input signal of $x[n]$ results in an output signal of $y[n]$, then an input of $k\,x[n]$ results in an output of $k\,y[n]$, for any input signal x and constant k. This kind of operation (multiplication by a scalar) is also called *scaling*.

- *Additivity*: If an input of $x_1[n]$ produces an output of $y_1[n]$, and a different input, $x_2[n]$, produces another output, $y_2[n]$, then the system is said to be additive, if an input of $x_1[n] + x_2[n]$ results in an output of $y_1[n] + y_2[n]$, for all possible input signals x_1, x_2. In words: signals added at the input produce signals that are added at the output.

- *Shift invariance*: A shift in the input signal will result in an identical shift in the output signal. In formal terms, if an input signal of $x[n]$ results in an output of $y[n]$, an input signal of $x[n+s]$ results in an output of $y[n+s]$, for any input signal and any constant, s.

Note: Shift invariance is not a strict requirement for linearity in mathematics, but it is a mandatory property for most DSP techniques. When you see the term 'linear system' used in DSP, you should assume it includes shift invariance unless you have reason to believe otherwise. Hence we will adopt this usage of the term 'linear system' as well.

Linear systems (2)

Linear systems have appealing mathematical properties. Let A, B be linear systems. Then the *composition* of A and B, defined by

- $(A \circ B)(x)[n] := A(B(x))[n]$,

is a linear system again. Moreover, if A, B, C are linear systems, then

- $(A \circ B)(x) = (B \circ A)(x)$ and (*commutativity*)
- $((A \circ B) \circ C)(x) = (A \circ (B \circ C))(x)$ (*associativity*)

for all signals x, where $\circ$ denotes the composition of linear systems. These properties follow immediately from the definitions. (*Exercise.*)

Linear systems (3)

Synthesis, decomposition and superposition

- *Synthesis*: Signals can be combined by scaling (multiplication of the signals by constants) followed by addition. The process of combining signals through scaling and addition is called synthesis.

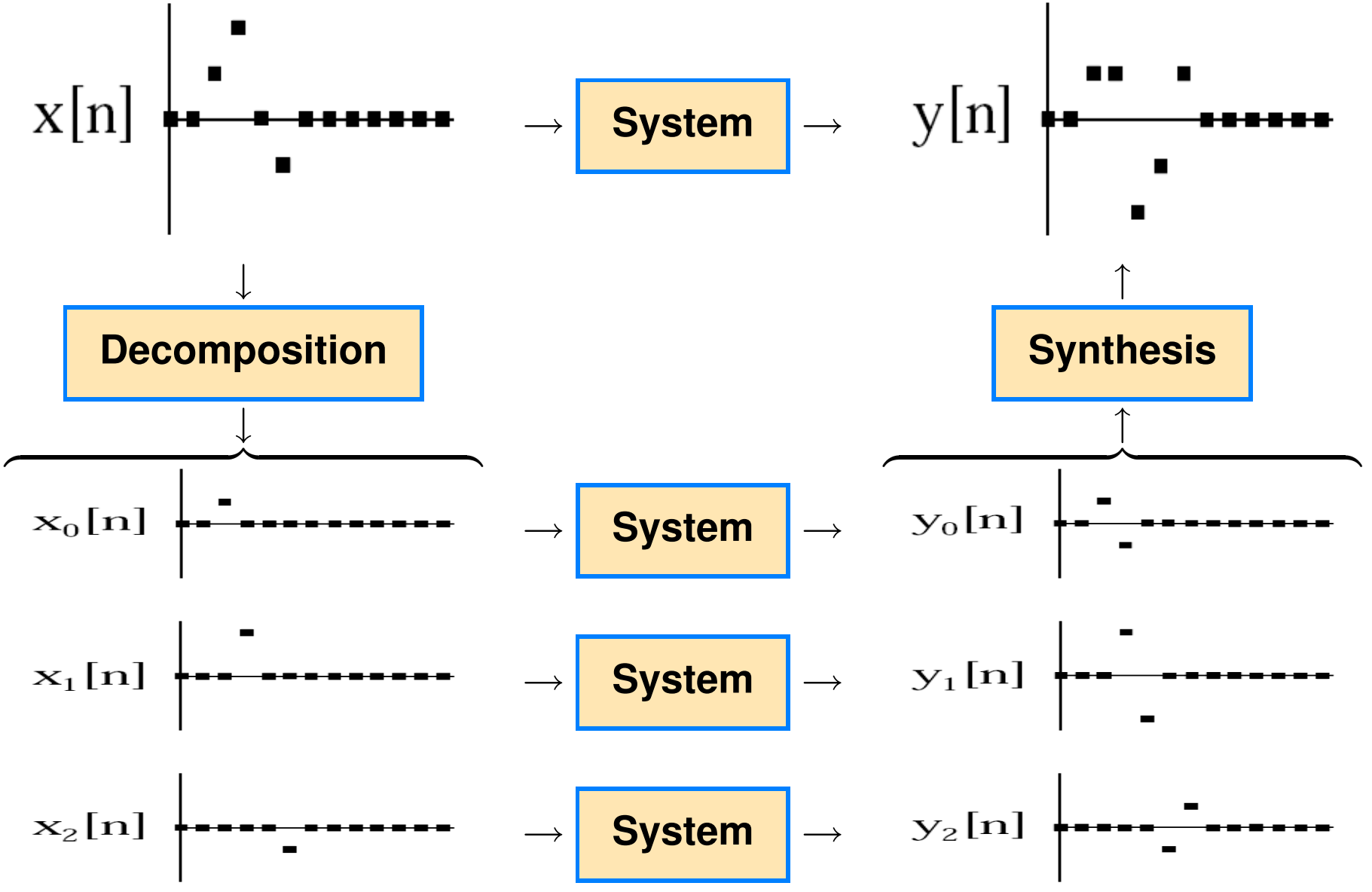
  For example, a mass spectrum can be composed out of baseline, a number of actual peaks, and white noise.

- *Decomposition* is the inverse operation of synthesis, where a single signal is broken into two or more additive components. This is more involved than synthesis, because there are infinite possible decompositions for any given signal.

Linear systems (4)

- *Superposition*: Consider an input signal, called $x[n]$, passing through a linear system, resulting in an output signal, $y[n]$.
  - Assume the input signal can be *decomposed* into a group of simpler signals: $x_0[n]$, $x_1[n]$, $x_2[n]$, etc. We will call these the *input signal components*.
  - Next, each input signal component is *individually* passed through the system,
  - resulting in a set of *output signal components*: $y_0[n]$, $y_1[n]$, $y_2[n]$, etc. These output signal components are then *synthesized* into the output signal, $y[n]$.

Linear systems (5)



Linear systems (6)

The trivial, but important observation here is:

> The output signal obtained by superposition of the components is *identical* to the one produced by directly passing the input signal through the system.

Thus instead of trying to understanding how complicated signals are changed by a system, all we need to know is how simple signals are modified.

There are two main ways to decompose signals in signal processing: *impulse decomposition* and *Fourier decomposition*.

Fourier methods

Fourier decomposition

The Fourier decomposition (named after Jean Baptiste Joseph Fourier (1768-1830), a French mathematician and physicist) uses *cosine* and *sine* functions as component signals:

$$c_k[i] := \cos(2\pi ki/N) \quad \text{and} \quad s_k[i] := \sin(2\pi ki/N)$$

where $k = 0, \ldots, N/2$ and N is the number of sample positions. (It turns out that $s_0 = s_{N/2} = 0$, so there are in fact only N components.)

Fourier based methods perform best for periodic signals. They are less suitable for peak-like signals, e. g. mass spectra or elution profiles. We will not go into further details here. Fourier theory *is* important in the analysis of FTICR mass spectrometers (FTICR = Fourier transform ion cyclotron resonance) and Orbitrap mass spectrometers.

Linear systems (2)

Impulse Decomposition

The impulse decomposition breaks an N samples signal into N component signals, each containing N samples. Each of the component signals contains one point from the original signal, with the remainder of the values being zero. Such a single nonzero point in a string of zeros is called an *impulse*.

The **delta function** is a normalized impulse. It is defined by

$$\delta[n] := \begin{cases} 1, & n = 0 \\ 0, & \text{otherwise.} \end{cases}$$

The **impulse response** of a linear system, usually denoted by $h[n]$, is the output of the system when the input is a delta function.

Linear systems (3)

How can we express the response of a linear system in closed form? Let us put together the pieces:

- The input signal can be decomposed into a set of impulses, each of which can be viewed as a scaled and shifted delta function.
- The output resulting from each impulse is a scaled and shifted version of the impulse response.
- The overall output signal can be found by adding these scaled and shifted impulse responses.

Thus if we know a system's impulse response, then we can calculate what the output will be for any possible input signal.

Linear systems (4)

We obtain the *standard equation for convolution*: If $x[n]$ is an N point signal running from 0 to $N-1$, and the impulse response $h[n]$ is an M point signal running from 0 to $M-1$, then the *convolution* of the two:

$$y[n] = x[n] * h[n]\,,$$

is an $N+M-1$ point signal running from 0 to $N+M-2$, given by:

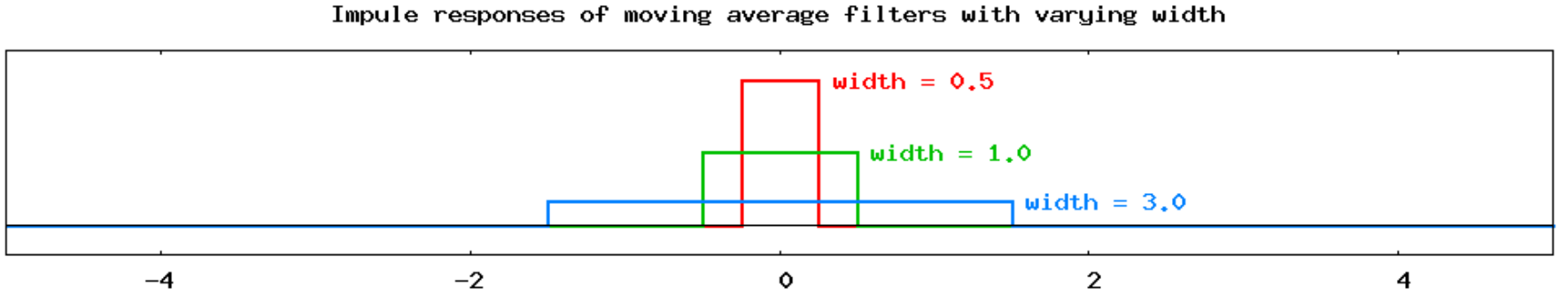
$$\boxed{y[i] = \sum_{j=0}^{M-1} h[j]\, x[i-j]\,.}$$

This equation is called the *convolution sum*.

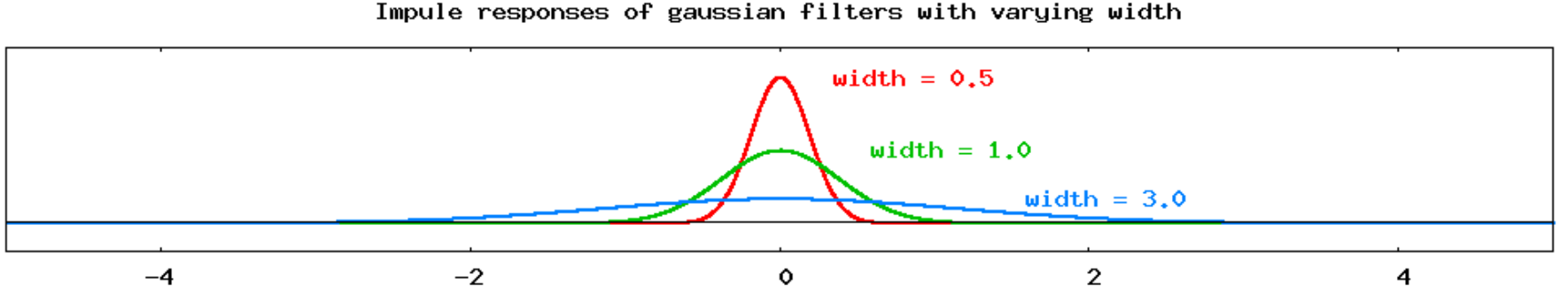
If the system being considered is a filter, then the impulse response is also called the *filter kernel*, the *convolution kernel*, or simply, the *kernel*. (In image processing, the impulse response is called the *point spread function*.)

Linear systems (5)

Moving average filter

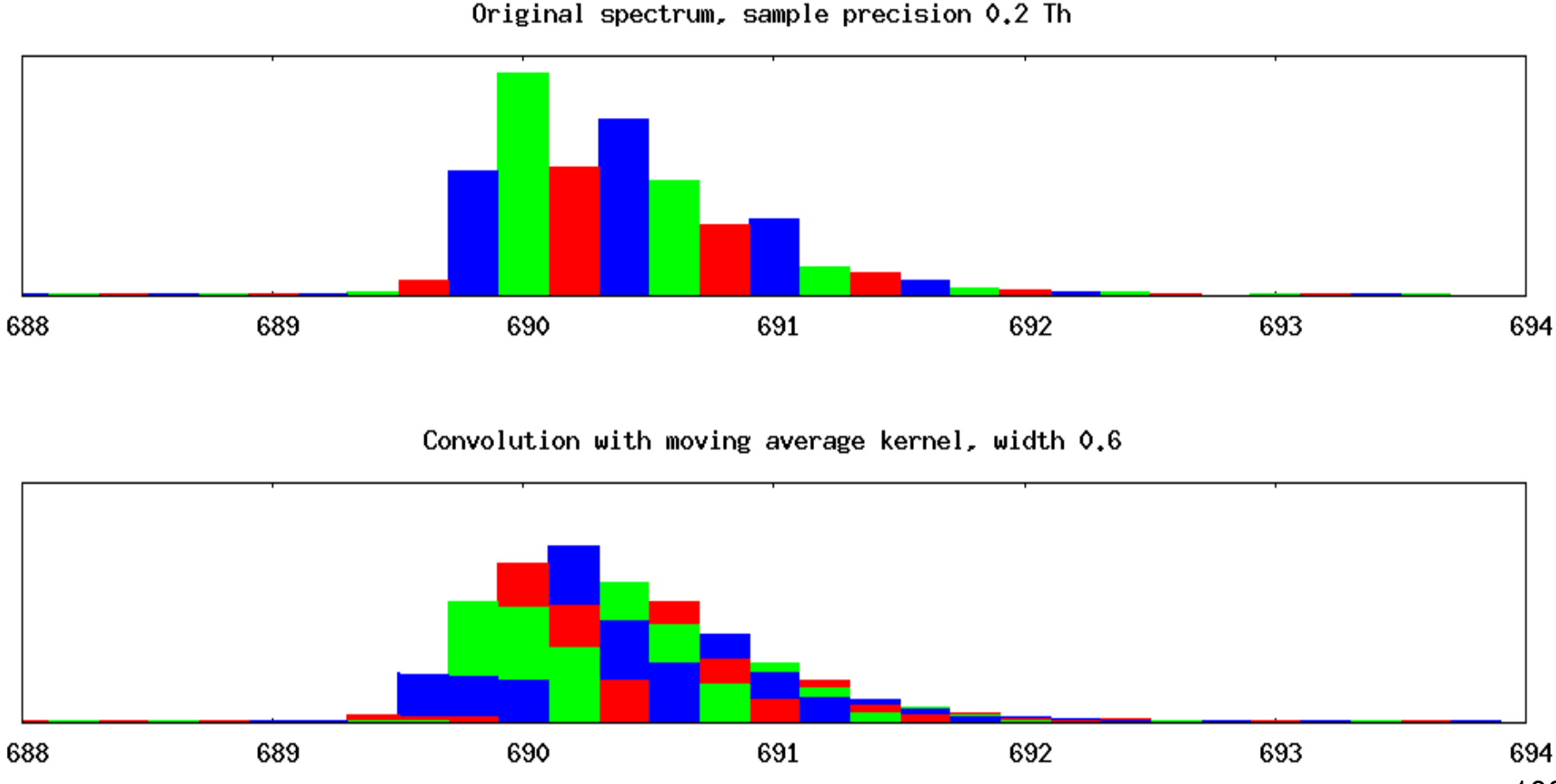


Gaussian filter



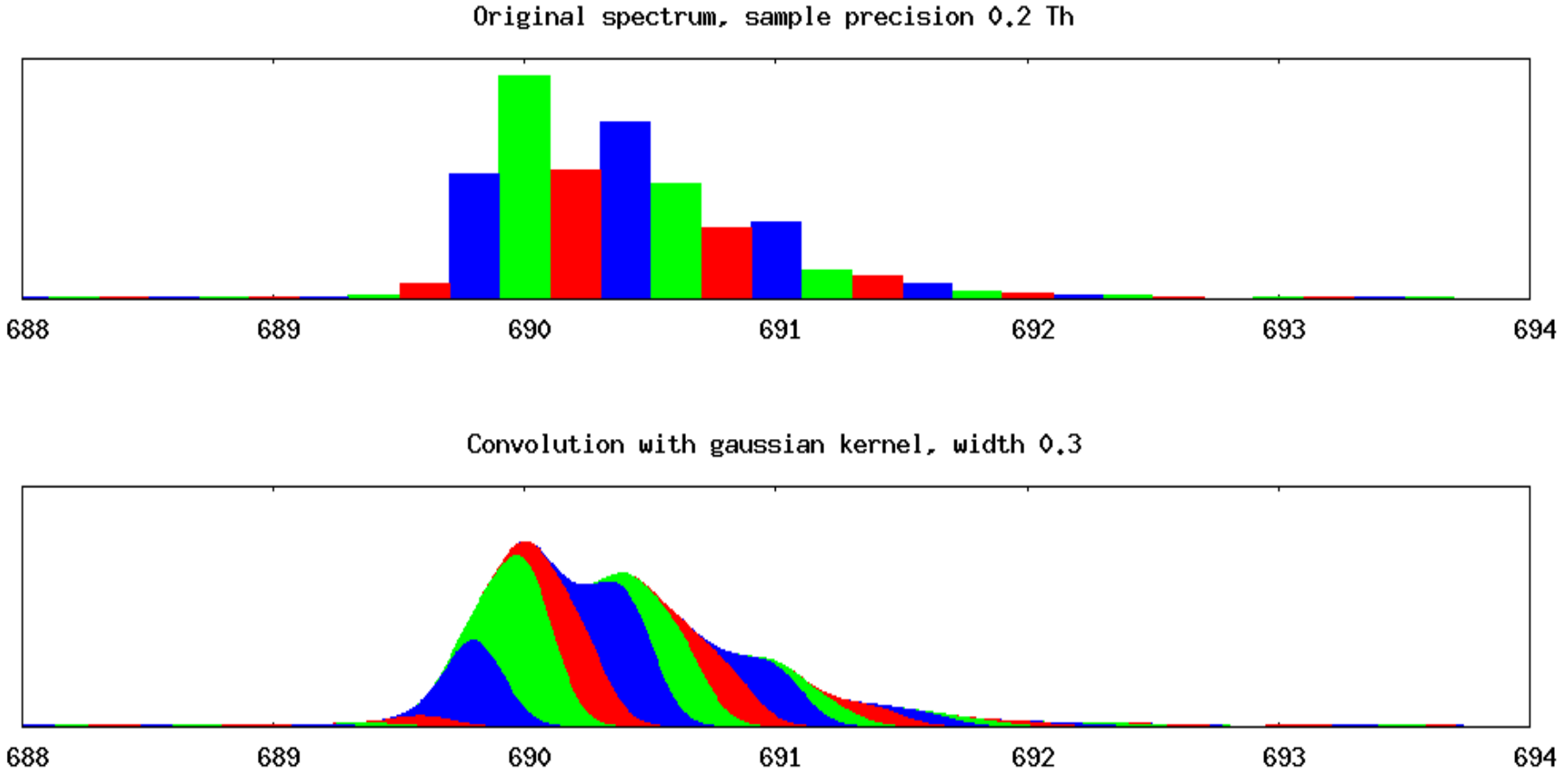
Linear systems (6)

The following illustration shows the contributions of the impulse responses in different colors. A moving average kernel was used that takes an unweighted average over three consecutive signal positions.



Linear systems (7)

And this shows the convolution with a Gaussian kernel.



Wavelets

So far we have only seen kernels without negative numbers. If we allow negative samples in the impulse response, we can construct linear filters that "accentuate" peaks instead of "dilating" them. This also leads us to the theory of *wavelets*.

A very commonly used wavelet is the *Marr wavelet*, also called *Mexican hat* for obvious reasons. It has the density function

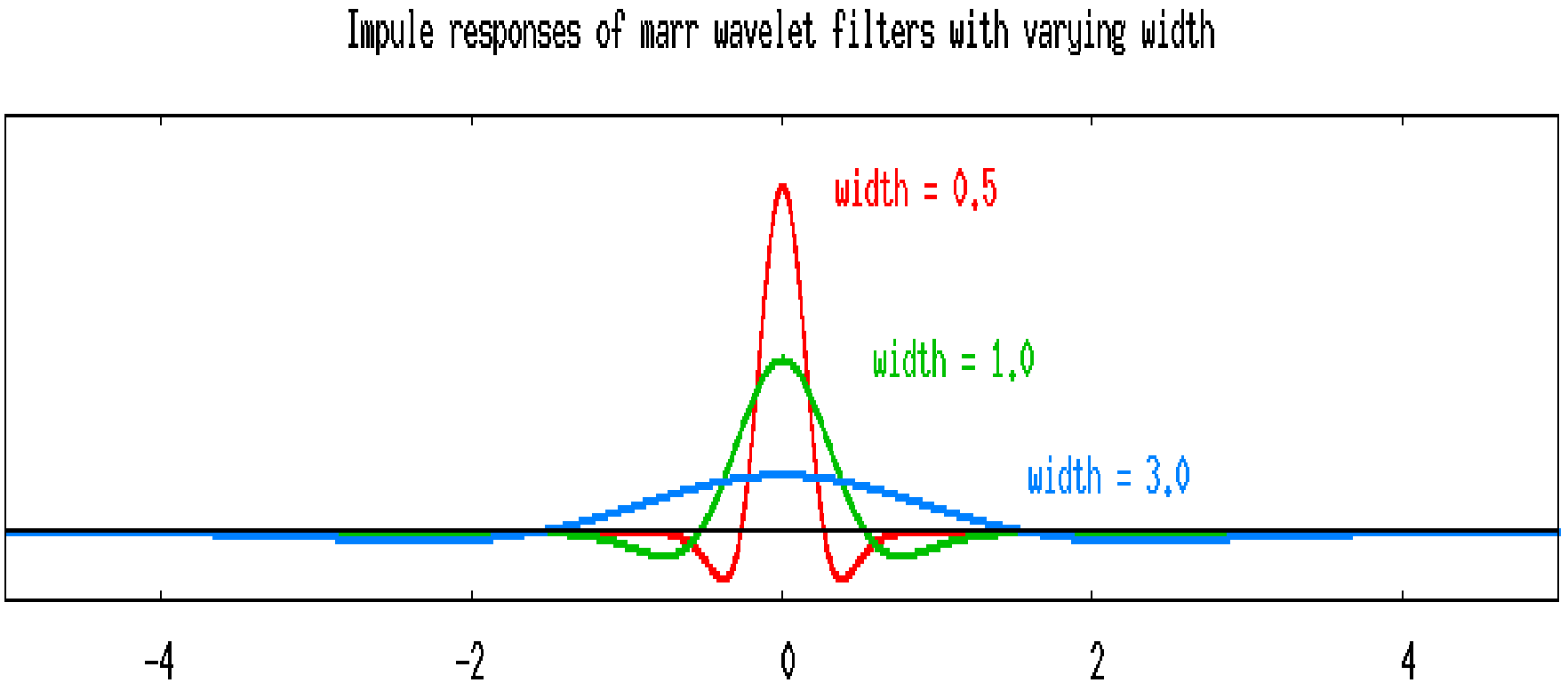
$$\psi(x) = (1 - x^2) \exp\left(\frac{x^2}{2}\right)$$

which is essentially the second derivative of the normal distribution:

$$\psi(x) = (1 - x^2) \exp\left(\frac{x^2}{2}\right) = \frac{d^2}{dx^2} \exp\left(-\frac{x^2}{2}\right)$$

Wavelets (2)

To adjust the width of the Marr wavelet, one introduces a scaling parameter a. Thus the impulse response would be $\psi(\frac{x}{a})/\sqrt{a}$.



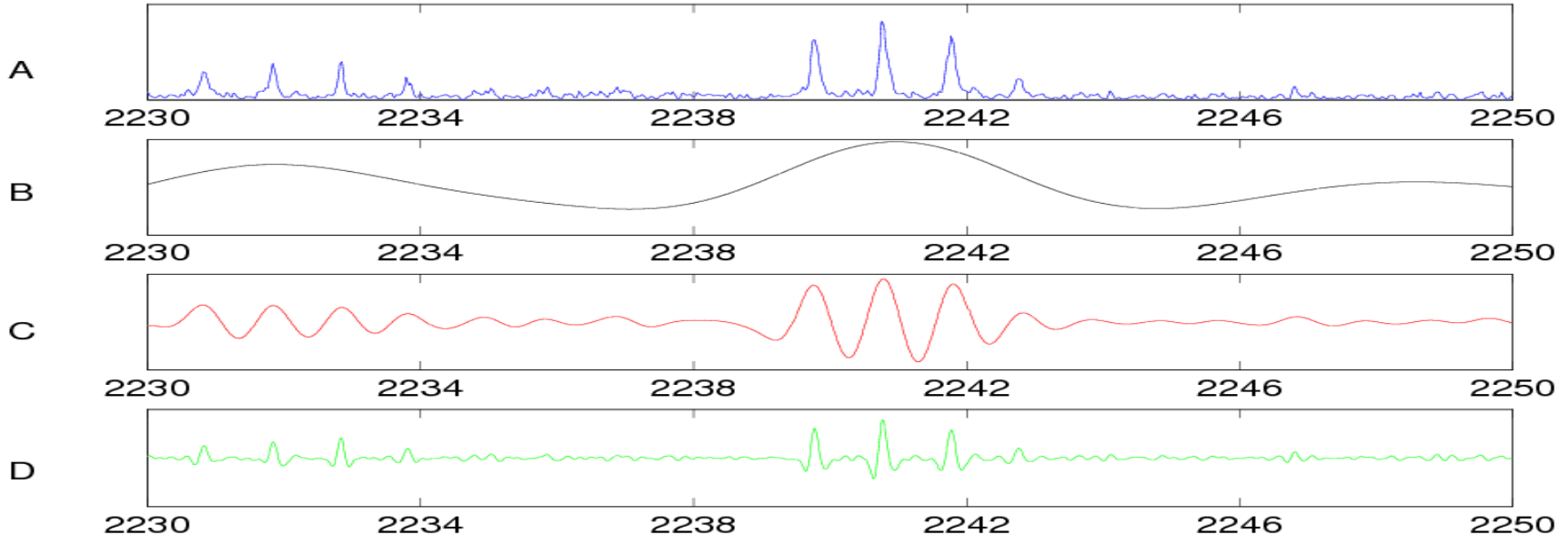
Wavelets (3)

Using wavelets, we can solve two problems at the same time:

- The integral of a wavelet is zero. Therefore a constant baseline has no influence on the output.
- High frequency noise is also filtered out.

Wavelets (4)

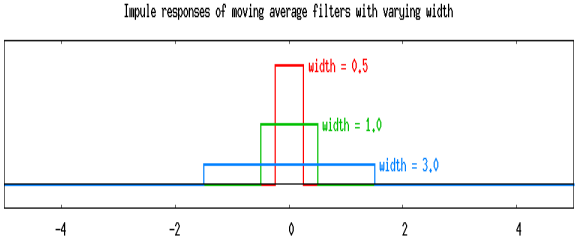
The following figures show that the Marr wavelet does a favorable job at detecting the maximum positions of the isotopic peaks.

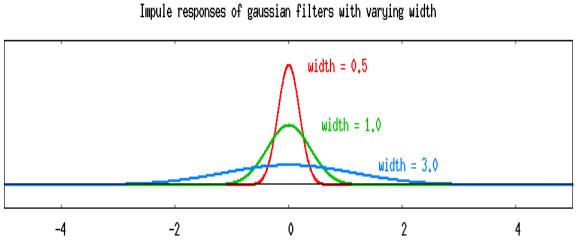


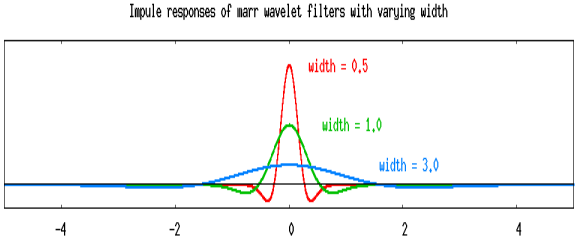
In the above plot A, B, C, and D the x-axis represents the mass interval between 2230Da and 2250Da, whereas the y-axis shows the intensity. A: Part of a MALDI mass spectrum. Plots B, C, and D show the continuous wavelet transform of the spectrum using a Marr wavelet with different dilation values a (B: $a = 3$, C: $a = 0.3$, D: $a = 0.06$).

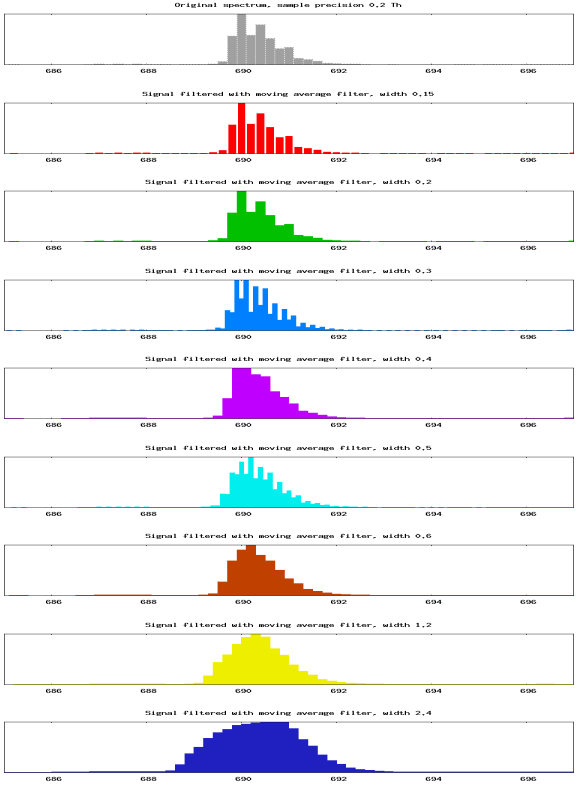
Wavelets (5)

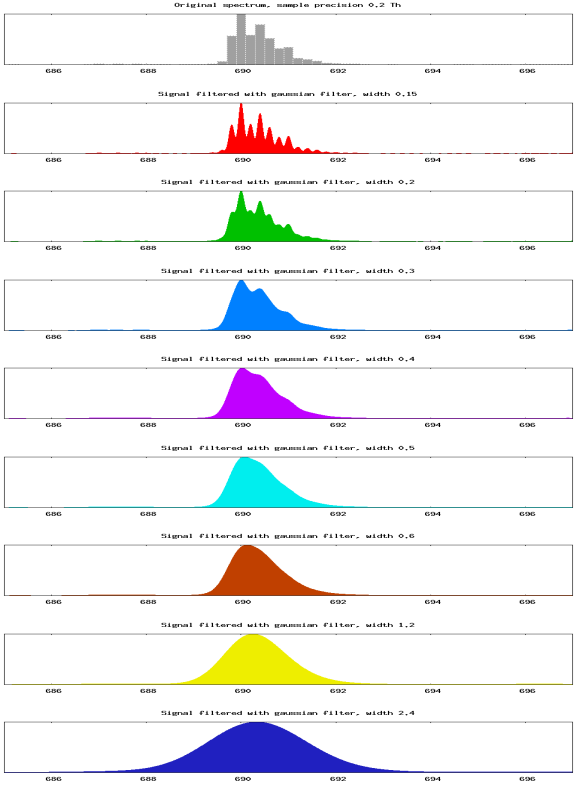
A comparison of filter kernels: Moving average, Gauss, Marr

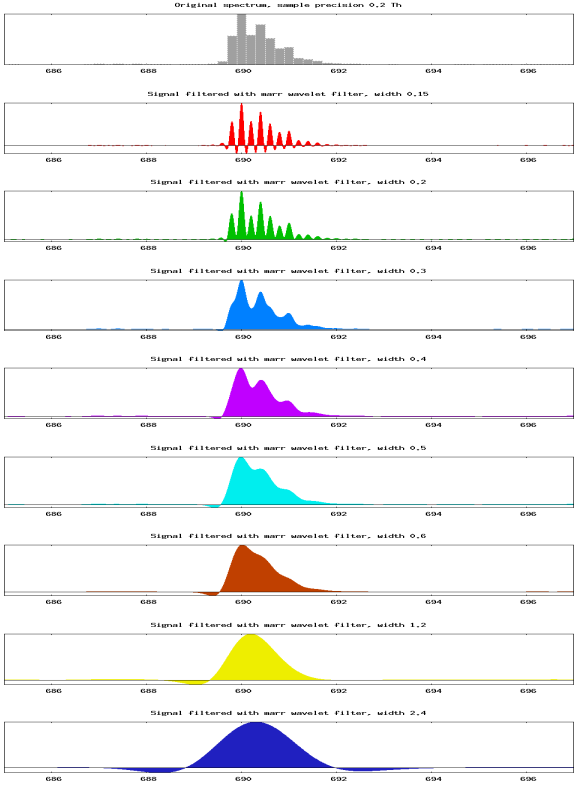






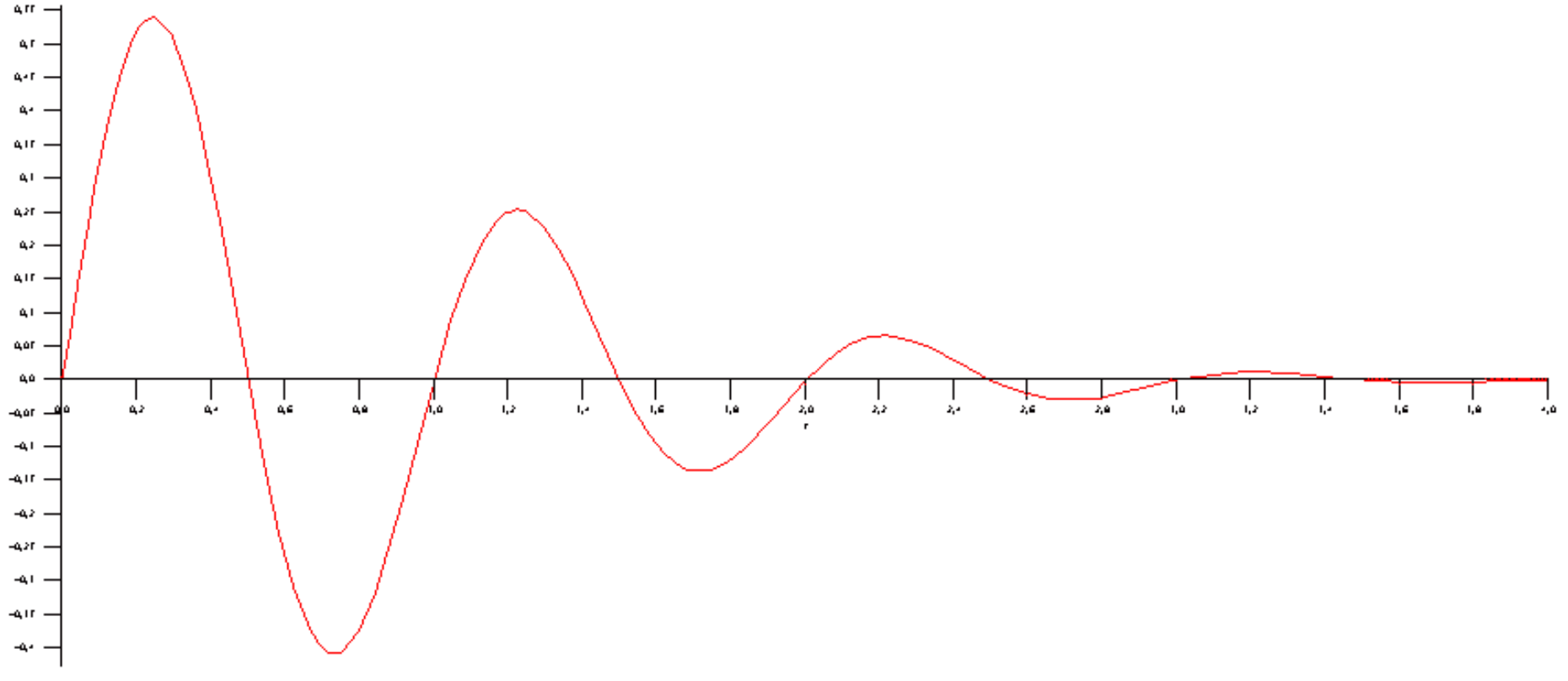






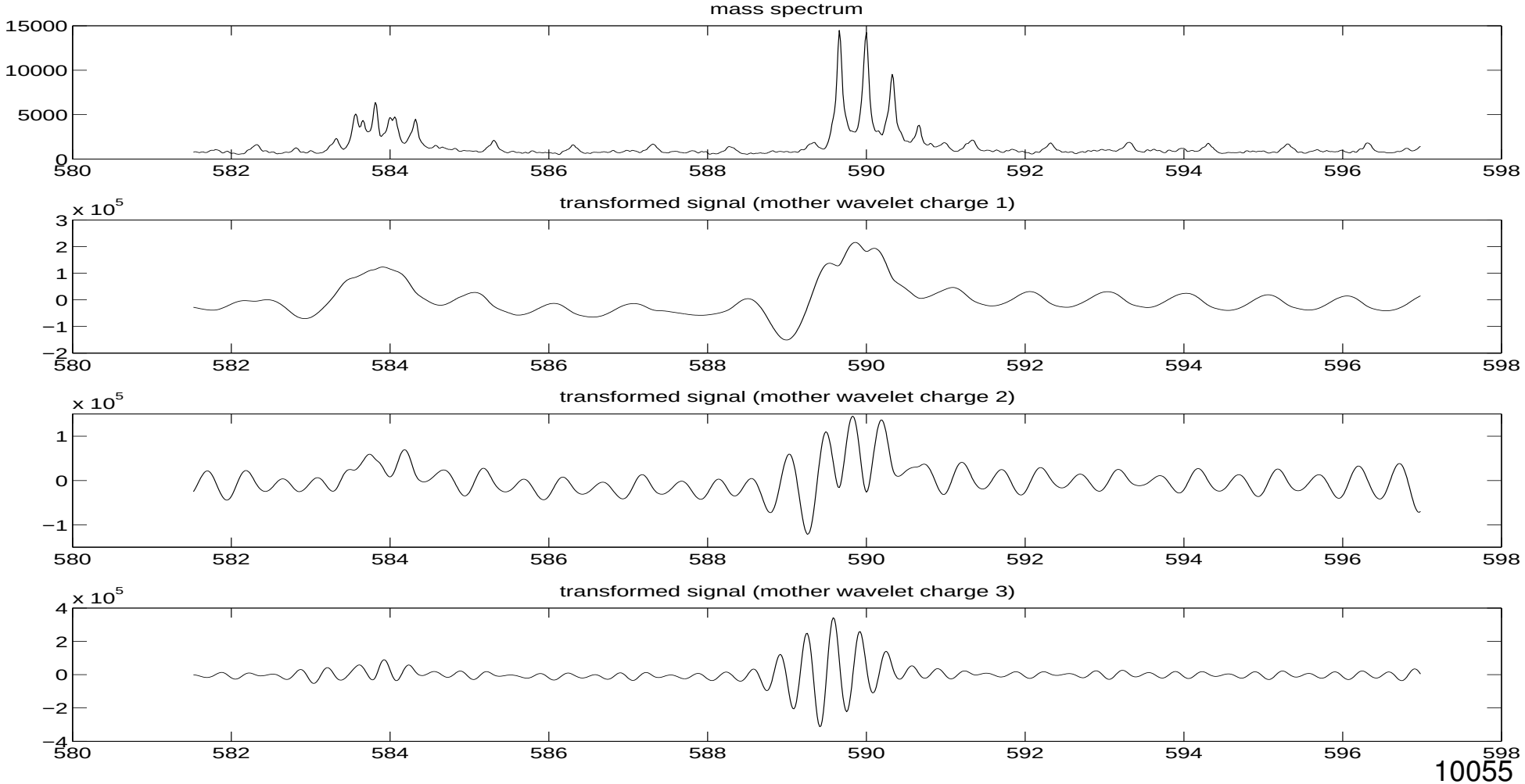
Wavelets (6)

Even better results can be obtained if we model the whole isotopic pattern by a wavelet:



Wavelets (7)

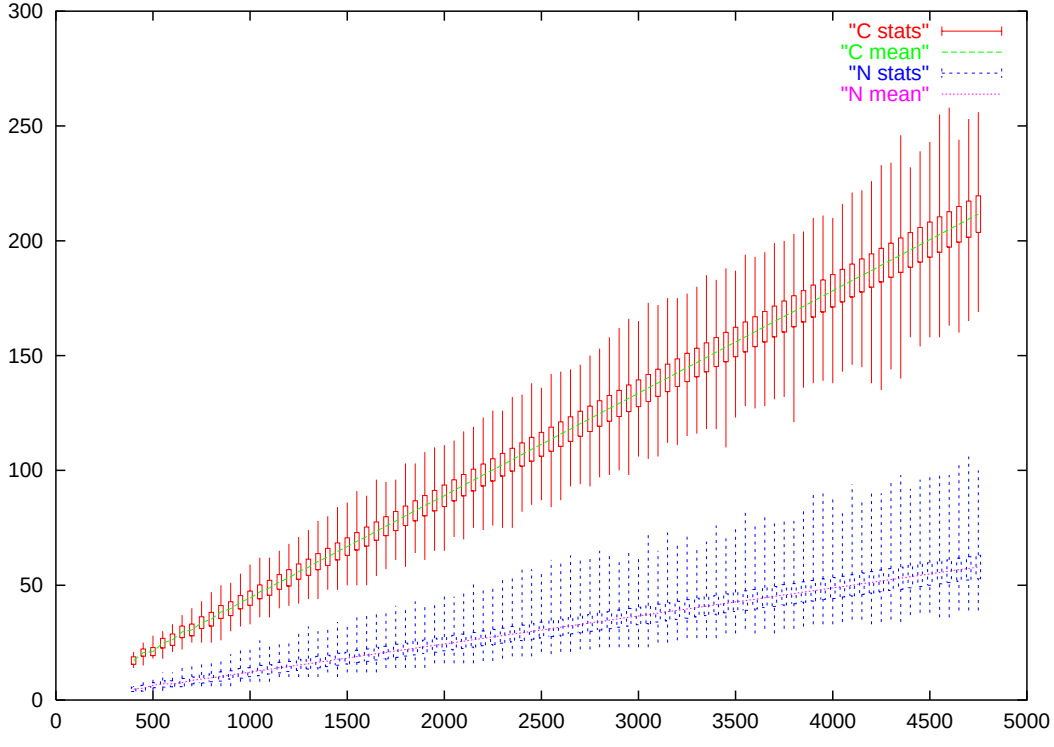
A mass spectrum and its isotope wavelet transform for charges one to three. The strong oscillating part of the transformed signal is a peptide of charge 3. The remaining peaks are non-peptidic compounds or noise.



Averagines

But how can we know the isotopic pattern to use for the wavelet?

If one plots the atomic content of proteins in some protein database (e.g. SwissProt) it becomes evident, that the number of atoms for each type grows roughly linearly. The picture shows on the x-axis the molecular weight and on the y-axis the number of atoms of a type.



Averagines (2)

Since the number of C,N, and O atoms grows about linearly with the mass of the molecule it is clear that the isotope pattern changes with mass.

mass	$\pi[0]$	$\pi[1]$	$\pi[2]$	$\pi[3]$	$\pi[4]$
1000	0.55	0.30	0.10	0.02	0.00
2000	0.30	0.33	0.21	0.09	0.03
3000	0.17	0.28	0.25	0.15	0.08
4000	0.09	0.20	0.24	0.19	0.12

- Since there is a very nice *linear relationship* between peptide mass and its atomic composition, we can estimate the average composition for peptide of a given mass.

- Given the atomic composition of a peptide, we can compute the relative intensities of its peaks in a mass spectrum, the *isotopic pattern*.

- We can use this knowledge for *feature detection* i.e. to summarize isotopic pattern into peptide features and to separate them from noise peaks.

Averagines (3)

Below is a spectrum with interleaved peptide peaks and noise. In this example, real peptide peaks are solid or dashed. Noise is dotted. Peaks 1 to 4 belong to two interleaved features of charge 1 with molecular weights of 1001 and 1001.5 Da, respectively.