Optimality property of the Gaussian window spectrogram

Citation for published version (APA):

DOI:
10.1109/78.80783

Document status and date:
Published: 01/01/1991

Document Version:
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:
- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license above, please follow below link for the End User Agreement:
www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:
openaccess@tue.nl
providing details and we will investigate your claim.
IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 39, NO. 1. JANUARY 1991

TABLE I

<table>
<thead>
<tr>
<th>Adaptation</th>
<th>Delay</th>
<th>Gain</th>
<th>Average Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20</td>
<td>0.498163</td>
<td>0.677243</td>
</tr>
<tr>
<td>2</td>
<td>15</td>
<td>0.447455</td>
<td>0.588629</td>
</tr>
<tr>
<td>3</td>
<td>18</td>
<td>-0.402277</td>
<td>0.513097</td>
</tr>
<tr>
<td>4</td>
<td>13</td>
<td>-0.367075</td>
<td>0.450902</td>
</tr>
<tr>
<td>5</td>
<td>12</td>
<td>-0.411588</td>
<td>0.393350</td>
</tr>
<tr>
<td>6</td>
<td>17</td>
<td>-0.390106</td>
<td>0.328782</td>
</tr>
<tr>
<td>7</td>
<td>23</td>
<td>-0.329475</td>
<td>0.282581</td>
</tr>
<tr>
<td>8</td>
<td>25</td>
<td>0.318380</td>
<td>0.227739</td>
</tr>
<tr>
<td>9</td>
<td>12</td>
<td>-0.228103</td>
<td>0.200542</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>0.246870</td>
<td>0.166498</td>
</tr>
</tbody>
</table>

Authorized licensed use limited to: Eindhoven University of Technology. Downloaded on March 05,2010 at 09:32:20 EST from IEEE Xplore. Restrictions apply.
over all normalized time-windows \( w(t) \) is achieved by the Gaussian window \( w(t) = 2^{1/4} \exp(-\pi t^2) \). Here, \((t_0, f_0)\) is the center of gravity of the signal \( x(t) \). \( S_{x,w}^{(w)}(t,f) \) is the spectrogram of \( x(t) \) due to the window \( w(t) \), and the double integral is a measure of the spread of \( S_{x}^{(w)}(t,f) \) around \((t_0, f_0)\) in the time-frequency plane.

I. INTRODUCTION

When \( x(t) \) is a square integrable signal, the spectrogram of \( x(t) \) due to \( w(t) \) is defined by

\[
S_{x,w}^{(w)}(t,f) = \int_{-\infty}^{\infty} x(\tau)w(\tau-t) e^{-2\pi i \tau f} d\tau
\]

(1.1)

or, equivalently, by

\[
S_{x,w}^{(w)}(t,f) = \int_{-\infty}^{\infty} X(\omega)W(\omega-f) e^{2\pi i \omega f} d\omega
\]

(1.2)

with \( X(f) \) and \( W(f) \) being the Fourier transforms of \( x(t) \) and \( w(t) \), respectively. We use here the definition

\[
y(f) = y(t) e^{-2\pi i f t} \]

(1.3)

for the Fourier transform \( Y(f) \) of \( y(t) \). In case \( x(t) \) is a signal whose energy is well concentrated around a point \((t_0, f_0)\) in the time-frequency plane, one would like \( S_{x,w}^{(w)}(t,f) \) to be well-concentrated around \((t_0, f_0)\) as well. As a measure for this, one could take

\[
\Sigma_{x,w}^{(w)}(t_0, f_0) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left[ (t-t_0)^2 + (f-f_0)^2 \right] S_{x,w}^{(w)}(t,f) \, dt \, df
\]

(1.4)

which is a quadratic time-frequency moment around \((t_0, f_0)\). In this correspondence we shall show the following. Assume that

\[
\int_{-\infty}^{\infty} \left| x(t) \right|^2 \, dt = \int_{-\infty}^{\infty} \left| X(f) \right|^2 \, df = 1
\]

(1.5)

and let

\[
t_0 = \int_{-\infty}^{\infty} t \left| x(t) \right|^2 \, dt, \quad f_0 = \int_{-\infty}^{\infty} f \left| X(f) \right|^2 \, df
\]

(1.6)

so that \((t_0, f_0)\) is the center of gravity of \( x(t) \). Then \( \Sigma_{x,w}^{(w)}(t_0, f_0) \) is uniquely minimized by the Gaussian

\[
w(t) = 2^{1/4} c \exp(-\pi t^2)
\]

(1.8)

with \( c \) a complex constant with \( |c| = 1 \).

This result is somewhat remarkable since the optimal \( w(t) \) is completely independent of \( x(t) \). We prove our result by relating \( \Sigma_{x,w}^{(w)}(t_0, f_0) \) to a different measure of concentration in the time-frequency plane, viz., to

\[
\sigma_{x}^2(\tau, g) = \int_{-\infty}^{\infty} \left( \tau - \tau_0 \right)^2 \left| y(\tau) \right|^2 \, d\tau
\]

\[
+ \int_{-\infty}^{\infty} \left( f - f_0 \right)^2 \left| Y(f) \right|^2 \, df
\]

(1.9)

where \( y \) is any signal and \((\tau, g)\) is any point in the time-frequency plane. We shall show that

\[
\Sigma_{x,w}^{(w)}(t_0, f_0) = \sigma_{x}^2(t_0, f_0) + \sigma_{x}^2(0, 0)
\]

(1.10)

so that minimization of \( \Sigma_{x,w}^{(w)}(t_0, f_0) \) reduces to minimization of \( \sigma_{x}^2(0, 0) \), and that \( \sigma_{x}^2(0, 0) \) is uniquely minimal for the Gaussian \( w(t) \) of (1.8).

II. DERIVATIONS

We shall prove formula (1.10) under the assumptions (1.5) and (1.7). We have from (1.4)

\[
\Sigma_{x,w}^{(w)}(t_0, f_0) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (t-t_0)^2 S_{x,w}^{(w)}(t,f) \, dt \, df
\]

(2.2)

Consider \( \Sigma_{x,w}^{(w)}(t_0) \). We have for any \( t' \)

\[
\int_{-\infty}^{\infty} S_{x,w}^{(w)}(t,f) \, df = \int_{-\infty}^{\infty} X(\omega)W(\omega-f) e^{2\pi i \omega f} d\omega
\]

(2.5)

Inserting (2.4) and (2.6) into (2.2) and (2.3), we get after a simple manipulation

\[
\Sigma_{x,w}^{(w)}(t_0) = \int_{-\infty}^{\infty} (t-t_0)^2 \left| x(\tau) \right|^2 \, d\tau + \int_{-\infty}^{\infty} \left| w(t) \right|^2 \, dt
\]

(2.8)

Here we have used (1.5) and (1.7), and the fact that

\[
\int_{-\infty}^{\infty} (t-t_0)^2 \left| x(\tau) \right|^2 \, d\tau = \int_{-\infty}^{\infty} (f-f_0)^2 \left| X(f) \right|^2 \, df = 0
\]

(2.9)

Adding (2.7) and (2.8) we readily obtain (1.10).

We next show that \( w(t) \) of (1.8) uniquely minimizes \( \sigma_{x}^2(0, 0) \). We have for any \( w(t) \)

\[
\sigma_{x}^2(0, 0) = \int_{-\infty}^{\infty} t^2 \left| w(t) \right|^2 \, dt + \int_{-\infty}^{\infty} f^2 \left| W(f) \right|^2 \, df
\]

\[
\geq 2 \left( \int_{-\infty}^{\infty} t^2 \left| w(t) \right|^2 \, dt \right)^{1/2} \left( \int_{-\infty}^{\infty} f^2 \left| W(f) \right|^2 \, df \right)^{1/2}
\]

\[
= \frac{1}{\pi}
\]

(2.10)

Formulas (2.4) and (2.6) were presented in [2, eq. (4.8) and (4.9)], without proof.
In the first inequality in (2.10) (which is the elementary inequality $a^2 + b^2 \geq 2ab$, $a > 0$, $b > 0$) we have equality if and only if $\sum_{i=1}^{n} f_i^2 |W(f)|^2 df = \sum_{i=1}^{n} f_i^2 |W(f)|^2 df$.

In the second inequality in (2.10) (which is the classical Heisenberg inequality, see [1, p. 273]) we have equality if and only if $w(t)$ is of the form $c(2\alpha)^{1/2} \exp(-\alpha t^2)$ for some $\alpha > 0$ and some complex $c$ with $|c| = 1$. Inserting this special form into (2.11), we readily obtain $\alpha = 1$, and the proof is complete.

**Acknowledgment**

The author is indebted to both referees who contributed considerably to a better presentation and who suggested important improvements and shortcuts in the author's original proof.

**References**


**Comparison of Algorithms for Standard Median Filtering**

Martti Juhola, Jyrki Katajainen, and Timo Raita

Abstract—In standard median filtering we search repeatedly for a median from a sample set which changes only slightly between the subsequent searches. We review several well-known methods for solving this running median problem, analyze the (asymptotical) time complexities of the methods, and propose simple variants which are especially suited for small sample sets, a frequent situation. Although we have restricted our discussion to the one-dimensional case, the ideas are easily extended to higher dimensions.

I. INTRODUCTION

A signal is here a sequence $S = (x_1, x_2, \cdots, x_n)$ of discrete values from a finite range. In median filtering [1]-[8] we slide a window of $m = 2k + 1$ samples and output $y_i = \text{median}(x_i, j = i - k, \cdots, i + k)$. The value $y_{i+1}$ is determined by rejecting the oldest sample $x_{i-k}$, inserting $x_{i+k+1}$, and searching the median again. We study data structures and algorithms for this running median problem. The view we have chosen rules out hardware-oriented techniques [6], [9], and [10], which are based on the bit representation of samples. Most methods can be extended to higher dimensions by replacing each sample by a vector of samples. We devise a data structure $W$ for the set $S$ supporting the following operations:

- create($S$) returns $W$ containing $x_1, x_2, \cdots, x_{2k+1}$,
- findmedian($W$) returns the median of the set of samples in $W$,
- replace($x$, $y$, $W$) removes the item $y$ from $W$ and adds $x$ into $W$.

To know $x$ and $y$ we must have a structure $Q$ for the time dependent operations:

- new$(t)$ returns the next sample value $x_{t+k+1}$,
- oldest$(t)$ returns the sample added to the set at time $t - k$.

Create is used only once in the initialization. The two others are applied once at each step. Therefore the total time complexity of the latter ones is important. Replace is time dependent and could be accomplished by:

- insert($x$, $W$) adds a new sample item $x$ into $W$,
- delete($y$, $W$) removes the item $y$ from $W$.

These operations imply that we should maintain the samples within a queue [11]-[14]. Often, however, the data structure which supports the time dependent operations is indistinguishable from the "main" structure $W$.

We can find efficient data structures for time independent findmedian, insert, and delete operations. For heaps [15] or search trees [14], each of them requires only $O(\log m)$ time. Thus they are theoretically optimal as to the asymptotic time complexity. Applying a sequence of $2k + 2$ (findmedian, replace) pairs we can sort a set $(x_1, \cdots, x_{2k+1})$ as follows. Let $-\infty \langle + \infty$ denote a dummy element smaller (respectively, larger) than any of $x_i$. Using these dummy elements we can sort the elements $x_i (i = 1, \cdots, k + 1)$ by applying findmedian and replace operations. First, we form a set of $2k + 1$ elements by performing $k$ (findmedian($W$), replace($-\infty$, $W$)) operation pairs followed by $k + 1$ (findmedian($W$), replace($x_i$, $W$)) pairs ($i = 1, \cdots, k + 1$) where the findmedians can be considered as empty. After this the $k + 1$ (findmedian($W$), replace($+\infty$, $W$)) operations give the elements $x_i$ in sorted order. By surrounding each block of $k + 1$ samples of the original signal with the dummy elements as described, we have a sequence the length of which is still $O(n)$. For sorting $k$ items we have an information theoretic lower bound $\Omega(k \log k)$. Thus, we have a lower bound $\Omega(n \log m)$ for filtering a signal of length $O(n)$ with a window size $m$. However, this is no more valid when the input is restricted to integers, since the complexity of algorithms is then expressed in $n$, $m$, and $U$, the size of integer universe. However, the performance of many algorithms is better than the theoretical results would suggest, since we can take advantage of the characteristics of the problem (e.g., the "continuity" of the signal).

II. ALTERNATIVE METHODS FOR IMPLEMENTING MEDIAN FILTERING

We shall present five ways to organize the data for efficient median filtering. We start with a structure in which no dependencies between the samples are stored. Then we gradually bring in facilities for characterizing more of the dependencies. In consequence, replace becomes slower and findmedian faster as we advance. The first two methods do not use sorting techniques in order to accomplish the median search [5], the next two methods do. In the last section we take advantage of the fact that the samples are integers from a given range.

All algorithms need a structure which holds the samples inside the window in time order ("age"). We use an array of length $m$ with a time pointer indicating the current oldest value. This is called a time ring (the oldest value is replaced at each step by a new value). The time independent operations are performed in structure $W$. Each slot of the ring contains a pointer to the corresponding item in $W$. 

1053-587X/91/0100-0204$10.00 © 1991 IEEE