A Framework for Evaluating the Quality of Lossy Image Compression

Jörg Haber       Hans-Peter Seidel

MPI–I–1999–4–001      December 1999
Author’s Address

Jörg Haber, Hans-Peter Seidel
Computer Graphics Group
Max-Planck-Institut für Informatik
Im Stadtwald, 66123 Saarbrücken, Germany
Email: {haberj, hpseidel}@mpi-sb.mpg.de
Abstract

In this research report we present a framework for evaluating and comparing the quality of various lossy image compression techniques based on a multiresolution decomposition of the image data. In contrast to many other publications, much attention is paid to the interdependencies of the individual steps of such compression techniques. In our result section we are able to show that it is quite worthwhile to fine-tune the parameters of every step to obtain an optimal interplay among them, which in turn leads to a higher reconstruction quality.

Keywords

image compression – wavelet transform – quantization – quality evaluation
1 Introduction

State-of-the-art algorithms for lossy still image compression are usually based on a multiresolution decomposition of the image data. The idea behind this technique has already been introduced in [BA83], the complete compression scheme was described in [ABDM92]. Due to a fast algorithm for a discrete wavelet transform published in [Mal89], this technique allows for high quality image compression in a fraction of the time needed for instance by fractal compression techniques such as [Jac92] or [Sau95].

The principle of such wavelet-based lossy image compression techniques can be grouped into three generic steps:

**compression**

1. basis transform of the image data;
2. quantization of the transformed coefficients;
3. coding of the quantized coefficients.

Each of these steps will be described in more detail in the following sections 2–4. Since usually steps 1 and 3 are losslessly invertible, the lossy compression happens in step 2.

The decompression is done in a straightforward way, which can be described again in three steps:

**decompression**

1. decoding of the compressed data;
2. dequantization of the decoded coefficients;
3. inverse basis transform of the dequantized coefficients.

Obviously, the second step will introduce some approximation error, since the quantization step in the compression scheme is not losslessly invertible. Thus, the approximated dequantized coefficients fed into the inverse basis transform step in the decompression scheme will be transformed into an approximation of the original image data. It is up to the design and implementation of every step and the interplay of all these steps to produce an approximation error as small as possible given a fixed compression ratio.

Even though a lot of work has been published on the examination of each step, cf. for example [OB96, HSW95, CS94, VBL95, CW90, Wic90, CW92] on step 1,
[Llo82, LBG80, Equ89, ABM91] on step 2 and [GG92, Sha93, VTC96] on step 3, there are only few publications to our knowledge that take into account at least some of the very important interdependencies of these three steps, cf. [Bal98]. Unfortunately, not all of these interdependencies can be motivated mathematically. In this case, an optimal set of parameters for each step has to be found heuristically. Evaluating all reasonable combinations of parameters can become a very time-consuming task. However, our results show that for a whole scenario of images, a set of sub-optimal parameters can be given, which produces an approximation error close to the optimal one in the average. The term *scenario of images* in our terminology denotes a collection of images, which are similar to each other by some means.

2 Transformation

The first step in the compression scheme is given by a basis transform of the image data. The objective of this transform is to reduce the entropy of the input data. Due to [Kar47, Loë48], the Karhunen-Loève transform is optimal in this sense. Unfortunately, the computational overhead for this transform is too big for practical applications. On the other hand, a (discrete) wavelet transform has proven to be a good compromise between computational time and entropy reduction. Particularly for use in lossy image compression, another transform has gained wide-spread popularity: the discrete cosine transform (DCT), as it is used e.g. in the JPEG standard [PM93]. However, because of the local support of the associated filter, a block structure artifact becomes clearly visible in the reconstructed images, especially when using high compression ratios. This makes the DCT rather unalluring for high quality image compression.

2.1 Wavelet Theory

In this section we will give a brief overview of wavelet theory as far as it applies to image compression. Good introductory textbooks on wavelet theory have been written by many authors, see for instance [Chu92, Dau92, Mey92, VK95, SN96]. Among the many approaches to wavelet theory the introduction of a multiresolution analysis (MRA) by Mallat [Mal89] seems to be most appropriate for image compression. An MRA of the \( L^2(\mathbb{R}) \) is a nested sequence of closed subspaces \( V_j \subset L^2(\mathbb{R}) \):

\[
\{0\} \subset \ldots \subset V_{-1} \subset V_0 \subset V_1 \subset \ldots \subset L^2(\mathbb{R}) \tag{1}
\]
such that
\[ \bigcup_{j \in \mathbb{Z}} V_j = L^2(\mathbb{R}) \quad (2) \]
\[ \bigcap_{j \in \mathbb{Z}} V_j = \{0\} \quad (3) \]
\[ f(\cdot) \in V_j \iff f(2^{-j} \cdot) \in V_0 \quad (4) \]
\[ \exists \varphi \in L^2(\mathbb{R}) \quad \text{with} \quad V_0 = \text{span}\{\varphi(\cdot - k) \mid k \in \mathbb{Z}\} \quad (5) \]

If in addition \( \{\varphi(\cdot - k) \mid k \in \mathbb{Z}\} \) is a Riesz basis of \( V_0 \) with some Riesz bounds \( 0 < A \leq B < \infty \), then it follows that for each \( j \in \mathbb{Z} \) the family \( \{\varphi_{j,k} \mid k \in \mathbb{Z}\} \) is a Riesz basis of \( V_j \), where
\[ \varphi_{j,k}(x) := 2^{j/2} \varphi(2^j x - k) \quad (6) \]

Such a function \( \varphi \in L^2(\mathbb{R}) \) is called a scaling function, if the subspaces
\[ V_j := \text{span}\{\varphi_{j,k} \mid k \in \mathbb{Z}\} \subset L^2(\mathbb{R}) \quad (j \in \mathbb{Z}) \]
satisfy the properties (1), (2) and (4).

A scaling function \( \varphi \) is said to generate an MRA. It can be shown that there exists a unique sequence \( \{h_k\} \ (h_k \in \mathbb{R}, k \in \mathbb{Z}) \) so that the so-called two-scale relation holds:
\[ \varphi(x) = \sqrt{2} \sum_k h_k \varphi(2x - k) \quad (7) \]

With the help of our function \( \varphi \) we can now define the associated (mother) wavelet
\[ \psi(x) := \sqrt{2} \sum_k g_k \varphi(2x - k) \quad (8) \]

with
\[ g_k := (-1)^k h_{1-k} \in \mathbb{R} \quad (9) \]
and, analogously to (6), its dilated and translated versions
\[ \psi_{j,k}(x) := 2^{j/2} \psi(2^j x - k) \quad (10) \]

Since \( \{\psi_{j,k} \mid k \in \mathbb{Z}\} \) is now an orthonormal basis of \( W_j \), where
\[ V_j \oplus W_j = V_{j+1} \quad (j \in \mathbb{Z}) \]
it follows that
\[ L^2(\mathbb{R}) = \bigoplus_j W_j , \quad (11) \]
i.e. \( \{ \psi_{j,k} \mid j, k \in \mathbb{Z} \} \) is an orthonormal basis of the \( L^2(\mathbb{R}) \). Thus any function \( f \in L^2(\mathbb{R}) \) can be represented as a linear combination of the basis functions \( \psi_{j,k} \):
\[ f(x) = \sum_{j,k} d_{j,k} \psi_{j,k}(x) . \quad (12) \]

2.2 Discrete Wavelet Transform

A multiresolution analysis of the \( L^2(\mathbb{R}) \) as it was introduced by MALLAT [Mal89] does not only lead to the construction of a wavelet \( \psi \), but in addition delivers an algorithm for a fast, discrete wavelet transform (DWT).

Let \( f \in V_0 \subset L^2(\mathbb{R}) \). According to (5), \( f \) can be written as
\[ f(x) = \sum_k c^0_k \varphi(x - k) \]
with real coefficients
\[ c^0_k := \langle f, \varphi_{0,k} \rangle_{L^2(\mathbb{R})} . \]

By using the notation
\[ c_k^m := \langle f, \varphi_{m,k} \rangle_{L^2(\mathbb{R})} , \quad d_k^m := \langle f, \psi_{m,k} \rangle_{L^2(\mathbb{R})} , \]
one yields with the help of the two-scale relation (7) the recursive identities:
\[ c_k^m = \sum_l h_l \langle f, \varphi_{m+1,2k+l} \rangle_{L^2(\mathbb{R})} = \sum_l h_l c_{2k+l}^{m+1} = \sum_l h_{l-2k} c_l^{m+1} , \]
\[ d_k^m = \sum_l g_l \langle f, \varphi_{m+1,2k+l} \rangle_{L^2(\mathbb{R})} = \sum_l g_l c_{2k+l}^{m+1} = \sum_l g_{l-2k} c_l^{m+1} . \]

These recursive formulas can be rewritten as
\[ c^m = H c^{m+1} , \quad d^m = G c^{m+1} , \quad (13) \]
The process of decomposing a signal \( c^0 \) into \( M \) levels of resolution is depicted in Figure 1. The inversion of this decomposition is called the reconstruction of \( c^0 \) (see Figure 2). It is given by

\[
c^{m+1} = H^*c^m + G^*d^m
\]  

with

\[
H^*c = \left\{ \sum_l h_{k-2l} c_l \right\}_{k \in \mathbb{Z}},
\]

\[
G^*c = \left\{ \sum_l g_{k-2l} c_l \right\}_{k \in \mathbb{Z}}.
\]
2.3 Two-Dimensional Wavelet Transform

A straightforward construction of a two-dimensional wavelet transform was proposed MALLAT [Mal89]. In the context of image compression, however, a tensor product wavelet transform is commonly used. If we define (analogously to Eqs. (14),(15)) new decomposition operators \( H_L, G_L \) and \( H_C, G_C \) that operate on the lines and on the columns of an image \( f \), respectively, the two-dimensional wavelet transform of \( f \) yields the following four sub-images:

\[
\begin{align*}
    f_{LL} &= H_C H_L f, \\
    f_{LH} &= G_C H_L f, \\
    f_{HL} &= H_C G_L f, \\
    f_{HH} &= G_C G_L f.
\end{align*}
\]  

(17)

The reconstruction of the image \( f \) is obtained by applying the corresponding adjoint operators \( H_C^*, G_C^*, H_L^*, G_L^* \) to these sub-images:

\[
\begin{align*}
    f &= H_L^* H_C^* f_{LL} + H_L^* G_C^* f_{LH} + G_L^* H_C^* f_{HL} + G_L^* G_C^* f_{HH} \\
    &= H_L^* [H_C^* f_{LL} + G_C^* f_{LH}] + G_L^* [H_C^* f_{HL} + G_C^* f_{HH}].
\end{align*}
\]  

(18)

MALLAT Algorithm

The MALLAT algorithm [Mal89] for a two-dimensional discrete wavelet transform with \( M \) levels of resolution is obtained by iterating the decomposition scheme (Eq. (17)) in the same way as it was shown for one-dimensional signals in Section 2.2. Using the notation from Section 2.3, the MALLAT algorithm with \( M \) levels of resolution can be written as:

\[
\begin{align*}
    f^{[0]} &\rightarrow \left\{ f^{[1]}_{LL}, f^{[1]}_{LH}, f^{[1]}_{HL}, f^{[1]}_{HH} \right\} \\
    &\rightarrow \left\{ f^{[2]}_{LL}, f^{[2]}_{LH}, f^{[2]}_{HL}, f^{[2]}_{HH}, f^{[1]}_{LL}, f^{[1]}_{LH}, f^{[1]}_{HL}, f^{[1]}_{HH} \right\} \\
    &\vdots \\
    &\rightarrow \left\{ f^{[M]}_{LL}, f^{[M]}_{LH}, f^{[M]}_{HL}, f^{[M]}_{HH}, \\
    &f^{[M-1]}_{LH}, f^{[M-1]}_{HL}, f^{[M-1]}_{HH}, \\
    &\vdots \\
    &f^{[1]}_{LL}, f^{[1]}_{LH}, f^{[1]}_{HL}, f^{[1]}_{HH} \right\}.
\end{align*}
\]

In this scheme, the upper index in parantheses denotes the level of resolution. The set of sub-images obtained by the MALLAT algorithm for \( M = 3 \) is shown in Figure 3(b).
Figure 3: Comparison of MALLAT Algorithm and Best-Basis Algorithm

Best-Basis Algorithm

In contrast to the MALLAT algorithm, the best-basis algorithm introduced by Coifman and Wickerhauser [CW92] performs a decomposition of every sub-image from the previous level of resolution to obtain the sub-images of the next level. From the (redundant) set of sub-images obtained by this decomposition process a subset is chosen subsequently. This subset is chosen to be both sufficient for lossless reconstruction of the original image and optimal in a certain sense. A common optimality condition is to minimize the sum of the pseudo-entropies of all sub-images. The pseudo-entropy of a sub-image \( s = (s_{i,j})_{i,j} \) is defined by:

\[
E(s) := - \sum_{i,j} s_{i,j}^2 \log_2 s_{i,j}^2
\]

with the predefinition \( 0 \cdot \log_2 0 := 0 \).

Starting with the sub-images \( f_{j,i}^{(M)} \) (\( j \in \{LL, LH, HL, HH\} := \{0, 1, 2, 3\}, i = 1, \ldots, 4^{M-1} \)) of the finest level \( m := M \), the sum of the pseudo-entropies of each quadrupel \( f_{j,i}^{(m)} \) (\( j = 0, \ldots, 3 \)) is compared to the pseudo-entropy of the corresponding sub-image \( f_{j,k}^{(m-1)} \) (\( j = i \mod 4, k = \lfloor \frac{i}{4} \rfloor \)) of the coarser level \( m-1 \). If

\[
\sum_{j=0}^{3} E(f_{j,i}^{(m)}) < E(f_{j,k}^{(m-1)}) ,
\]

the four sub-images \( f_{j,i}^{(m)} \) (\( j = 0, \ldots, 3 \)) are kept and the sum of their pseudo-entropies is assigned to the sub-image \( f_{j,k}^{(m-1)} \), otherwise the four sub-images of level \( m \) are discarded. This process is carried out for all levels \( m = M, \ldots, 1 \).
In the end, a subset of sub-images remains, that fulfils both the reconstruction and the optimality condition. A possible set of sub-images generated by the best-basis algorithm for $M = 3$ is shown in Figure 3(c).

3 Quantization

After a wavelet transform has been applied to the image data, the resulting real wavelet coefficients are mapped to some integer symbols. It is obvious, that in general this step cannot be reverted without introducing some approximation error. Thus the design of the quantization step plays an important role for the reconstruction quality. A scalar quantization quantizes every single wavelet coefficient seperately, a vector quantization groups several wavelet coefficients together and maps this coefficient vector to a single output symbol. The latter is usually more efficient, but also computationally more expensive.

3.1 Scalar Quantization

A uniform scalar quantization can be obtained by equidistant partitioning of the interval of the real coefficients into $n$ subintervals. These subintervals (quantization intervals) are numbered from $0, \ldots, n-1$, and for every wavelet coefficient the number of its subinterval is output to the coder. Because of the following coding step, $n$ should generally be chosen to be a power of 2. During the dequantization step, the midpoint of every quantization interval is used as an approximation for all the coefficients in this interval. This choice is known to minimize the average approximation error in an $L^2$ norm, cf. [Say96, GG92].

Since wavelet coefficients have an expected value of $\mu = 0$, it is advantageous to use an odd number (e.g. $n = 2^p - 1$) of quantization intervals lying symmetrically around zero. Thus the approximation value 0.0 will be used for all wavelet coefficients lying close to zero.

The approximation error can be further reduced by adapting the partitioning of the coefficient interval to the frequency distribution of the wavelet coefficients. This technique is called a nonuniform scalar quantization or pdf-quantization. A finer partitioning resulting in a smaller approximation error is used in regions with many wavelet coefficients while a coarser partitioning increases the approximation error for regions with only few wavelet coefficients.

1 except those from the smooth residuum of the last transform step
frequency distribution of the wavelet coefficients:

\[ f(x) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}. \]  

(20)

Since \( \mu = 0 \), the standard deviation \( \sigma \) of the coefficients is the only parameter to be fitted.

LLOYD and MAX [Llo57, Max60] have introduced an algorithm to adapt the partitioning of the coefficient interval to the model (20). Again, the (nonuniform) quantization intervals are numbered from 0, \ldots, \( n-1 \) and these numbers are output as symbols to the coder. Note that for the dequantization step the standard deviation \( \sigma \) has to be stored in addition to the boundary values of the coefficient interval and the number \( n \) of subintervals.

### 3.2 Vector Quantization

Unlike the scalar case, a vector quantizer groups several wavelet coefficients together and outputs a single symbol for this coefficient vector. On the one hand, this can greatly reduce the number of different symbols needed, on the other hand, the approximation error might become too large. It is the crucial point of every vector quantizer to use a carefully chosen set of code vectors, which act as representatives for the coefficient vectors. For every coefficient vector, its best matching code vector is determined, and the number of this code vector is output to the coder. In the dequantization step, a code vector is used as an approximation for all the coefficient vectors which have been mapped to this code vector. The best matching code vector for a particular coefficient vector is the one which minimizes the Euclidian distance to the coefficient vector among all code vectors.

The most important aspect of vector quantization is the computation of the codebook, i.e. the set of all code vectors. This task has been studied by LINDE, BUZO and GRAY [LBG80]. They proposed an iterative algorithm called the LBG algorithm, which converges against a local optimal codebook. An even better adapted codebook can be obtained by using our modified LBG algorithm, see [HS00].

### 4 Coding

The final step in the compression scheme takes the symbols from the quantizer and codes them into a bitstream, which can be written to a file. The coding itself is losslessly invertible. Many different coding schemes have been developed and studied, ranging from a fast but usually not very efficient run-length coding over a variable code size Huffman coding [Huf52] to any of the variants of a quite efficient but computationally expensive arithmetic coding [Pas76, RL79,
Instead of re-describing these well-known coding techniques, we refer to the literature mentioned before or to the excellent textbook by GERSHO and GRAY [GG92], which covers these aspects as well.

One aspect of coding deserves special attention: more efficient coding schemes like Huffman coding or arithmetic coding use a variable code size for each input symbol depending on the symbol’s probability of occurrence. However, the latter is unknown to the coder a priori. Therefore the coder has to adapt to the probabilities of occurrence with every new symbol being coded. Again we refer to [GG92] for a detailed description of this adaptive control.

5 Implementation

Our framework for evaluating the quality of lossy image compression consists of several independent modules, which can be easily exchanged or extended. The following list gives a brief overview of the capabilities of our system:

- **wavelet transform:**
  - many different pre-defined wavelets + user-definable wavelets
  - MALLAT algorithm
  - best-basis algorithm

- **quantization:**
  - uniform scalar quantization
  - non-uniform scalar quantization
  - vector quantization

- **coding:**
  - adaptive arithmetic coding
  - adaptive arithmetic coding with preceeding run-length coding

The coefficients of the wavelets available in our framework are taken from [Dau92, SS96, BCR91, Dau93, ASH87, ABDM92, VBL95, VTC96, OB96, Bri93]. Additional wavelets can be integrated easily by the user. Both wavelet transforms are implemented as described in Section 2.3.

For the quantization step we implemented the techniques from Section 3.1 and 3.2. In order to find an optimal bit allocation for the quantization of the subbands [JN84, GS88], we use a numerical bisection method to minimize a cost functional subject to auxiliary conditions (LAGRANGE method).
Our coding scheme is restricted to an adaptive arithmetic coder [BCW90], which has been modified to be optionally preceded by a run-length coding step. The arithmetic coding is performed in integer arithmetics, the adaption of the coding to the input data follows the technique presented in [GG92, Sec. 9.7].

6 Results

One major aspect that influenced the development of our framework was to compare different parameter settings for each of the compression steps and to study their interdependencies. We chose several digitized images from both the Hubble Space Telescope Archive [HST99] and the FBI fingerprint archive [Bri96]. These two archives are denoted as scenarios in the following. Some of the chosen test images are shown in Figure 4 and Figure 5. For each of the test images we performed numerous compression–decompression–comparison cycles using different parameter settings each time. The following parameters have been varied according to the given ranges:

1. choice of the wavelet: 18 different wavelets have been examined; the most useful wavelets for image compression according to our tests are printed in Table 1 (page 15);
2. choice of the transform algorithm (MALLAT vs. best-basis);
3. number of levels of resolution ($M = 3, \ldots, 8$);
4. choice of quantization technique: uniform scalar, non-uniform scalar, and vector quantization; using vector quantization, the codebook size $n = 2^p$, $p = 7, \ldots, 10$ and the code vector length $l = 4, 16$ have been varied as well (see [HS00] for details);
5. choice of coding scheme: adaptive arithmetic coder with or without preceeded run-length coding.

All combinations of settings for these parameters have been evaluated and analyzed. The decompressed images have been compared to the original images using both the well-know peak-signal-to-noise ratio (PSNR) and the distortion measure adapted to human perception (DMHP) proposed in [BWW96]. A detailed description of the results has been published in [Hab99]. In this report we will give a short summary of the results obtained. It is worthwhile to note that these results differ only very little between different images within each scenario. Therefore we present the results grouped by the two scenarios investigated.
6.1 Astronomy Scenario

The results of our simulations using images from the astronomy scenario can be summarized as follows:

- very good results are obtained by using the wavelets $\mathcal{W}_1$, $\mathcal{W}_2$ and the MAL-LAT algorithm with $M = 5, \ldots, 8$ levels of resolution;

- using the best-basis algorithm with $M = 5, \ldots, 7$, the wavelet $\mathcal{W}_1$ produces very good results as well;

- uniform scalar quantization produces better results than non-uniform scalar quantization (see remark in Section 6.3);

- vector quantization yields outstanding results compared to scalar quantization: a code vector length of $l = 16$ should be chosen always, a codebook size of $n \geq 256$ is usually sufficient to produce better results than a scalar quantization (cf. Section 6.3);
an additional run-length coding step before the arithmetic coding leads to very little improvement of the results.

### 6.2 Fingerprint Scenario

For the fingerprint scenario, the following observations have been made:

- the wavelet $\mathcal{W}_1$ yields very good results using the MALLAT algorithm and $M = 4, \ldots , 7$ levels of resolution;
- even better results are produced by using the wavelets $\mathcal{W}_1, \mathcal{W}_3$ and the best-basis algorithm with $M = 4, \ldots , 6$;
- concerning the quantization step, the same observations as stated in Section 6.1 have been made; the only difference is that a code vector length of $l = 4$ yields better results for small codebook sizes $n \leq 256$ in this scenario;
- the benefit of an additional run-length coding step before the arithmetic coding depends heavily on the quantization technique being used: it is recommended to use a run-length coding in combination with a uniform quantization; there is no improvement when using vector quantization.

### 6.3 Summary

Our simulations showed that it is possible to find a common set of parameters for each of the examined scenarios, which leads to very good results in the average when compressing arbitrary images from these scenarios. We suggest to run a similar parameter analysis in all cases, where a huge number of images from a single scenario has to be compressed.

However, some observations from our simulations need to be explained in more detail. At first glance, one expects from a non-uniform scalar quantizer to produce better results than a uniform scalar quantizer. Our results showed correctly, that this is generally not the case. The reason for this unexpected behaviour is given by the entropy coding step that follows the quantization step. A non-uniform quantizer generates symbols whose probabilities of occurrence are rather uniformly distributed. This is actually the worst case for any entropy coding! To our knowledge, the only published statement about this fact is given in [GG92, Sec. 9.9]: “... a uniform quantizer is approximately optimal if entropy coding is used.”

Vector quantization has proven to be a very powerful technique in our simulations. Even though the computational costs for compressing images can become quite high, the advantage of fast decompression and high reconstruction quality (given a fixed compression ratio) cannot be neglected. Our simulations show, that
one codebook is sufficient for all images from one scenario. Such a scenario-based codebook can be computed once and does not have to be stored with the compressed data. In this case, a codebook size of $n = 1024$ produces very good results.

7 Conclusions and Future Work

In this research report we describe a framework for determining the quality of lossy image compression techniques based on a wavelet decomposition of the image data. Our framework offers several possibilities to perform each of the three steps “transform — quantization — coding” characteristic for this compression technique.

From our simulations we can conclude that it is indeed worth the effort to perform an analysis on several images of a scenario and find a (scenario dependent) set of compression parameters, if a large number of images from a single scenario has to be compressed subject to a high reconstruction quality.

It is left as a future work task to develop techniques and algorithms that are able to find a (sub-)optimal set of compression parameters without evaluating all possible parameter combinations. Such techniques might use available image information like for instance color / gray tone histograms or frequency distributions obtained through a Fast Fourier Transform to restrict the space of parameter combinations being evaluated.
Table 1: Wavelet coefficients (orthogonal wavelets have identical coefficients for analysis (A) and synthesis (S), biorthogonal wavelets have different coefficients for each)

<table>
<thead>
<tr>
<th>name</th>
<th>mask coefficients</th>
<th>reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{W}_1$</td>
<td>A + S: { $-3.793512864280802 \cdot 10^{-3}$, $7.78296425672740 \cdot 10^{-3}$, $2.345269614207717 \cdot 10^{-2}$, $-6.5771912814936 \cdot 10^{-2}$, $-6.1123900297255 \cdot 10^{-2}$, $4.05176902491182 \cdot 10^{-1}$, $7.93777226260872 \cdot 10^{-1}$, $4.284834763773700 \cdot 10^{-1}$, $-7.179982161915484 \cdot 10^{-2}$, $-8.230192710629983 \cdot 10^{-2}$, $3.45502757329774 \cdot 10^{-2}$, $1.588054486366945 \cdot 10^{-2}$, $-9.007976136730624 \cdot 10^{-3}$, $-2.574517688136797 \cdot 10^{-3}$, $1.117518770830630 \cdot 10^{-3}$, $4.662169598204029 \cdot 10^{-4}$, $-7.098330250637900 \cdot 10^{-5}$, $-3.459977319727278 \cdot 10^{-5}$ }</td>
<td>[BCR91, Dau93]</td>
</tr>
<tr>
<td>$\mathcal{W}_2$</td>
<td>A: { $3.78284550699546 \cdot 10^{-2}$, $-2.38494650193800 \cdot 10^{-2}$, $-1.106244044184234 \cdot 10^{-1}$, $3.774028556126538 \cdot 10^{-1}$, $8.526986790094034 \cdot 10^{-1}$, $3.774028556126538 \cdot 10^{-1}$, $-1.106244044184234 \cdot 10^{-1}$, $-2.38494650193800 \cdot 10^{-2}$, $3.78284550699546 \cdot 10^{-2}$ }</td>
<td>[ABDM92], [VBL95]</td>
</tr>
<tr>
<td></td>
<td>S: { $-6.453888262893844 \cdot 10^{-2}$, $-4.068941760955844 \cdot 10^{-2}$, $4.18092273222122 \cdot 10^{-1}$, $7.884856164056644 \cdot 10^{-1}$, $4.18092273222122 \cdot 10^{-1}$, $-4.068941760955844 \cdot 10^{-2}$, $-6.453888262893844 \cdot 10^{-2}$ }</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>mask coefficients</td>
<td>reference</td>
</tr>
<tr>
<td>------</td>
<td>-------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>$\mathcal{W}_3$</td>
<td>A: ${ 2.885256501123136 \times 10^{-2}, \ 8.244478227504624 \times 10^{-5}, \ -1.575264469076351 \times 10^{-1}, \ 7.679048884691436 \times 10^{-2}, \ 7.589077294537619 \times 10^{-1}, \ 7.589077294537619 \times 10^{-1}, \ 7.679048884691436 \times 10^{-2}, \ -1.575264469076351 \times 10^{-1}, \ 8.244478227504624 \times 10^{-5}, \ 2.885256501123136 \times 10^{-2} }$</td>
<td>[VTC96]</td>
</tr>
</tbody>
</table>

**References**


Below you find a list of the most recent technical reports of the Max-Planck-Institut für Informatik. They are available by anonymous ftp from ftp.mpi-sb.mpg.de under the directory pub/papers/reports. Most of the reports are also accessible via WWW using the URL http://www.mpi-sb.mpg.de. If you have any questions concerning ftp or WWW access, please contact reports@mpi-sb.mpg.de. Paper copies (which are not necessarily free of charge) can be ordered either by regular mail or by e-mail at the address below.

Max-Planck-Institut für Informatik
Library
attn. Anja Becker
Im Stadtwald
66123 Saarbrücken
GERMANY
e-mail: library@mpi-sb.mpg.de

MPI-I-1999-3-005 T.A. Henzinger, J. Raskin, P. Schobbens  Axioms for Real-Time Logics
MPI-I-1999-3-004 J. Raskin, P. Schobbens  Proving a conjecture of Andreeka on temporal logic
MPI-I-1999-3-002 J. Raskin, P. Schobbens  The Logic of Event Clocks
MPI-I-1999-2-008 A. Bockmayr, F. Eisenbrand  Cutting Planes and the Elementary Closure in Fixed Dimension
MPI-I-1999-2-007 G. Delzanno, J. Raskin  Symbolic Representation of Upward-closed Sets
MPI-I-1999-2-006 A. Nonnengart  A Deductive Model Checking Approach for Hybrid Systems
MPI-I-1999-2-005 J. Wu  Symmetries in Logic Programs
MPI-I-1999-2-004 V. Cortier, H. Ganzinger, F. Jacquemard, M. Veanes  Decidable fragments of simultaneous rigid reachability
MPI-I-1999-2-003 U. Waldmann  Cancellative Superposition Decides the Theory of Divisible Torsion-Free Abelian Groups
MPI-I-1999-2-001 W. Charatonik  Automata on DAG Representations of Finite Trees
MPI-I-1999-1-007 C. Burnikel, K. Mehlhorn, M. Seel  A simple way to recognize a correct Voronoi diagram of line segments
MPI-I-1999-1-006 M. Nissen  Integration of Graph Iterators into LEDA
MPI-I-1999-1-005 J.F. Sibeyn  Ultimate Parallel List Ranking?
MPI-I-1999-1-004 M. Nissen, K. Weihe  How generic language extensions enable “open-world” desing in Java
MPI-I-1999-1-003 P. Sanders, S. Egner, J. Korst  Fast Concurrent Access to Parallel Disks
MPI-I-1999-1-001 A. Crauser, P. Ferragina  A Theoretical and Experimental Study on the Construction of Suffix Arrays in External Memory
MPI-I-98-2-018 F. Eisenbrand  A Note on the Membership Problem for the First Elementary Closure of a Polyhedron
MPI-I-98-2-017 M. Tzakova, P. Blackburn  Hybirdizing Concept Languages
MPI-I-98-2-014 Y. Gurevich, M. Veanes  Partisan Corroboration, and Shifted Pairing
MPI-I-98-2-013 H. Ganzinger, F. Jacquemard, M. Veanes  Rigid Reachability