

# **Time-Variant Watermarks for Digital Videos: An MPEG-Based Approach**

Ernst L. Leiss  
University of Houston

## **Abstract**

Watermarks provide a means of embedding information into digital videos that can be used for a variety of purposes, such as establishing ownership, tracing origin of copies, etc. We outline an approach that permits a significant increase of the amount of information that can be accommodated in a watermark, namely time-variant watermarks. The approach is formulated assuming video represented in an MPEG format.

Implementation issues of time-variant watermarks are discussed, as are their advantages over the usual time-invariant watermarks, with emphasis on defeating attacks using filtering, cropping, resizing, and other standard methods used to defeat watermarks, such as changing existing frames, as well as new attacks, such as removing or adding frames.

# **When Good Algorithms Yield Bad Programs**

Ernst L. Leiss  
University of Houston

## **Abstract**

Disciplined software development typically starts with one or more suitable algorithms that are then translated into code. There exists a well-developed body of theory related to the question what constitutes a good algorithm. Apart from the obvious requirement of correctness, the most important quality of an algorithm is its efficiency. Computational complexity provides the tools for determining the efficiency of an algorithm; often, it is relatively easy to capture the efficiency of an algorithm in this way. However, for the software developer the ultimate goal is efficient software, not efficient algorithms. It is often not well understood how to go from a good algorithm to good software. It is this transition that the talk addresses.

Computational complexity is based on fundamental assumptions which help designers in analyzing algorithms. Unfortunately, many of these assumptions are violated by modern computing environments. As a result, it is quite possible to start with a good algorithm and end up with bad software, either altogether incorrect or of unacceptable performance.

We examine the fundamental assumptions of algorithm analysis and explain, with concrete examples, how these assumptions may fail to hold for software. We explore consequences and implications of these differences and outline techniques that avoid the resulting pitfalls. The overarching objective is achieving the desired goal of producing software as efficient and effective as possible.

## Short Biography

Ernst Leiss is Professor of Computer Science at the University of Houston. He earned degrees in computer science (M. Math., University of Waterloo, Canada, 1974), engineering (Dipl.-Ing., TU Vienna, Austria, 1975) and mathematics (Dr. techn., TU Vienna, Austria, 1976). He taught at Waterloo, the University of Chile in Santiago, and the University of Kentucky before joining the faculty at the University of Houston in 1979. His research interests include formal language theory, high-performance computing, and security. He has supervised and conducted research resulting in 17 doctoral dissertations, over 100 master theses, six books, and about 170 peer-reviewed publications in conferences and journals.

Leiss has lectured in 32 different countries, in three different languages. He has been an ACM Distinguished Lecturer since 1991 and the chair of the Houston chapter of the IEEE Computer Society since 1981.