COMPRESSION OF SMIL DOCUMENTS

Chia–Yuan (Andy) Teng
Nokia Mobile Phones
12278 Scripps Summit Dr. San Diego, CA 92131 (e–mail: andy.teng@nokia.com)

The World Wide Web Consortium (W3C) standard Synchronized Multimedia Integration Language (SMIL, pronounced as ”smile”) is an HTML–like mark–up language to describe temporal behavior and presentation layout for multimedia objects. SMIL is widely used in today’s internet. Many video clips send SMIL documents to clients before transmitting video and audio streams.

Data compression is a process to reduce the number of bits in a representation of data. It can save storage space and reduce bandwidth requirement. Data compression technologies may be classified into two categories: lossless and lossy. The compression technology that can recover data perfectly from the compressed bits is called lossless compression. The one that cannot is called lossy compression. There are many lossless data compression algorithms, such as Ziv–Lempel methods, have been developed in recent years. However, the performance to compress the SMIL document with such methods is not very impressive because they don’t exploit the special format of the SMIL document. As a result, the rate to compress a SMIL document is almost the same as that to compress English text, i.e. 2–4 bits/char. For achieving a better compression, a method based on a new parsing technology will be proposed.

The proposed compression algorithm contains two major procedures: parsing and coding. Parsing is a procedure to segment a SMIL document into non–overlapping phrases (a string of consecutive characters). Coding is a procedure to assign a codeword to the phrase. Both the phrase and codeword are of variable length.

There are many ways to do text parsing. The method proposed here is to parse the SMIL document into variable–length phrases based on the boundaries of SMIL entities (elements, tags, attributes, AttValue, etc.). Specifically, the SMIL document is parsed by the following hierarchical procedure

1. Parse a SMIL document into elements.
2. Parse an empty element into (1) ’<’ and element name, (2) attributes and (3) ’/>’. If some attributes are required attributes (attributes definitely appear with their associated element), merge them with (1).
3. Parse a non–empty element into (1) STag, (2) child elements and (3) ETag.
4. Parse a STag into (1) ’<’ and element name, (2) attributes and (3) ’>’.
5. Parse an attribute into (1) attribute name and ’=’, and (2) AttValues.
6. Parse an AttValue into phrases or symbols. For instance, the AttValue of ”src” is segmented into (1) protocol name (e.g. http://www. or rtsp://www.), (2) a string of alphanumerals and (3) extension (e.g. .html or .jpg).

After parsing, a variable–length code is assigned to each phrase based on the conditional probability.

We apply the proposed algorithm on several SMIL documents. As a result, its performance is 400%–600% better than other commercial compression methods, such as COMPRESS and GZIP.