# Multiresolution Document Analysis with Wavelets

Amen Zwa David S. Ebert Ethan L. Miller

November 22, 1996

**Abstract**

The $n$-gram analysis technique breaks up a text document into several $n$-character long unique grams, and produces a vector whose components are the counts of these grams. A typical corpus contains hundreds of thousands of such grams. Wavelet compression reduces the dimension of the $n$-gram vectors, and speeds up document query operations. Document vectors with their dimensions reduced to four components is readily represented in a three dimensional volume.

**Keywords:** information retrieval, $n$-gram analysis, wavelets, multiresolution analysis, volume visualization, glyph-based volume rendering.

## 1 Introduction

Text documents are represented as multi-dimensional vectors in $n$-gram document analysis technique. The document is broken down into unique *grams*. The frequencies of these grams are used as the components of the vector representing that document. A gram is a consecutive sequence of $n$ characters that appear in the document. The vector representation allows us to use the cosine of the angle between the two vectors as a similarity measure between the two documents.

The use of wavelets have multiplied in recent years: signal processing, noise reduction, image compression, to name a few. Here, we discuss the use of wavelets to speed up the document queries. We also present a glyph-based volume visualization of document clusters, and conclude by enumerating future extensions to our approach.

## 2 $n$-gram and Wavelet Analysis of Documents

The $n$-gram analyzer breaks up a text document into several $n$-character long grams. For instance, the word `wavelet` will form `wavel`, `avele`, etc., 5-grams. The output of the analyzer is an $n$-gram vector whose components are the counts of these grams in that document. A document with $m$ such unique grams has a corresponding vector representation in $m$-dimensional vector space. Alternatively, this vector can be viewed as a one-dimensional signal with $m$ samples. When converting a corpus of text files to $n$-gram vectors, the analyzer constructs a *centroid* vector whose components are the averages of the gram counts across the entire corpus [4]. We now briefly describe how wavelet decomposition can speed up the $n$-gram-based document similarity computations. A paper by Strang [6] provides a more detailed introduction to wavelets.

A one-dimensional signal, such as the $n$-gram vector above, with $m = 2^j$ samples is an element in a vector space $V^j$ [5]. For example, let the vector space $V^0$ be the space of all constant functions over the interval $[0, 1)$, and $V^1$ the space of all functions with two equal constant pieces. Similarly, $V^j$ is the space of all functions with $2^j$ equal constant pieces. This nested space $V^j$ is essential to the multiresolution analysis (MRA) [3] of signals. The inner product $< u \mid v > = \|u\|\|v\|cos\theta$ is typically used to determine the orthogonality of the two vectors $u$ and $v$. In our document analysis, $u$ and $v$ are the $n$-gram vectors in the space $V^j$. The $cos\theta$ measures the *similarity* between the two documents.

The wavelet transform decomposes the input signal into a low frequency *smooth* half and a high frequency *detail* half. At each level of decomposition, we are able to reduce the effective size of the vector to half that of the previous level by eliminating all the detail coefficients from the similarity calculations.

When the document retrieval engine receives a user query, it calculates the similarity between the query and every document in the corpus. The similarity between the query $q$ and the document $d$ with $m$ unique $n$-grams is calculated as $sim(q, d) = \sum_{i=1}^{m} q_i d_i / [(\sum_{i=1}^{m} q_i^2)^{\frac{1}{2}} (\sum_{i=1}^{m} d_i^2)^{\frac{1}{2}}]$. The time to compute the three $m$-iteration loops in the formula grows linearly with the number of grams and with the number of documents in the corpus. It is common for large corpra to contain hundreds of thousands of unique $n$-grams among tens of thousands of documents. Repeated wavelet discompositions of the document vector from $V^j$ to $V^{j-k}$ reduces the size of the vector to $1/2^{j-k}$ of the original. The similarity computation time is therefore reduced from $m$ iterations per loop to $m/2^{j-k}$.

In general, the accuracy of the similarity value decreases as the compression level $j$ increases. We may exploit the multi-resolution nature of wavelet decomposition as follows. First, we calculate the similarities of the query against the entire corpus at some high compression level. Though imprecise, this *rough* comparison does narrow down the search field. We then perform *fine* comparisons of the query against the new, but smaller, search set using the original uncompressed vectors. This idea may be extended to the distributed corpra. The brokers perform rough comparisons on compressed vectors to determine the corpus in which the desired document may reside. The backend search engine associated with that corpus then performs fine comparisons on uncompressed vectors to obtain the exact similarity values. We may further improve the performance by gradually increasing the resolution. At each level, we compute similarities with twice the resolution, but against a smaller search set than the previous level.

# 3 Visualization of Documents

We display document vectors in a three dimensional volume by reducing their dimension to four components. It is not supperising that at such extreme compression levels, the wavelet transform preserves little infomation of the document that the resulting similarity values are virtually useless. However, these four coefficients are enough to cluster documents spatially.

We map the first three components to $(x, y, z)$ position of a document *glyph*, and the fourth to its color. A glyph is a three dimensional geometric object that represents an entity. For instance, a small cube can indicate the position of a document within the visualization volume. Similar documents are close to each other in space, and share the same hue. Documents belonging to the same topic form a cloud of certain color. By assigning the cube representing the centroid a different color, we can percieve the spatial relations of the document clusters with respect to the central *theme* of the corpus. We can extend this technique to eight dimensions (the next finer resolution level) by mapping the higher four dimensions to shape, size, opacity, and orientation of the glyph. But, it quickly becomes difficult to comprehand the full meaning of these attributes as the number of dimensions grow [1].

Beshers and Feiner [1] describe a system that uses the *Worlds within Worlds* metaphor to visualize interactively the multivariate functions. For an $m$-dimensional function $f(x_1, x_2, \ldots, x_m)$, the first three parameters $(x_1, x_2, x_3)$ place the first *inner world* within the *outter world*; the second set of parameters $(x_4, x_5, x_6)$ place the second inner world within the first inner world; and, so on. The innermost world displays the function as a surface. There are a total of $\lceil m/3 \rceil$ worlds. However, it appears that eight dimensions is at the limit of the usefulness of this approach.

The $2\frac{1}{2}$-dimensional surfaces employed by *Worlds within Worlds* technique does not suite our needs due to the high number of dimensions in our data. We can, however, extend it to full three-dimensional volume as follows. First, scale the lengths of all the $n$-gram vectors by dividing them with the maximum length, making the longest vector in the corpus a unit vector. Next, offset each vector component triplets from the previous set along the volume axes. Finally, place a document glyph at the current offset when there remains two or less vector dimensions. Map the remaining dimensions to size and color of the glyph. These modifications allow a true volume visualization of documents in arbitrary dimensional vector space.

## 4 Future Work

The conventional $n$-gram analysis converts text documents into multi-dimensional vectors, and uses the inner product to calculate similarities between a pair of documents. We mentioned above that a one-dimensional signal with $2^j$ samples can be represented in a vector space $V^j$. A text file is indeed a stream of bytes that can be treated as a one-dimensional signal. Hence, it is possible to apply the wavelet transform on the text file. We plan to experiment with the application of $n$-gram analysis to the transformed documents. We also plan to use $D_4$ four-coefficient Daubechies wavelets [2] in place of Haar.

## References


[1] C. Beshers and S. Feiner, AutoVisual: Rule-Based Design of Interactive Multivariate Visualization, *IEEE Computer Graphics and Applications*, 13(4), July 1993, 41-49.

[2] I. Daubechies, Orthonormal Bases of Compactly Supported Wavelets. *Communications on Pure and Applied Mathematics*, (41)7, October 1988, 909-996.

[3] S. Mallat, A Theory for Multiresolution Signal Decomposition: The Wavelet Representation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (11)7, July 1989, 674-693.

[4] C. Pearce, Telltale Document Retrieval System, *Ph.D. Dissertation*, Computer Science and Electrical Engineering Department, University of Maryland Baltimore County, 1994.

[5] E. Stollnitz, T. DeRose, and D. Salesin, Wavelets for Computer Graphics: A Primer, *IEEE Computer Graphics and Applications*, (15)3, May 1995, 76-84 (Part 1) and (15)4, July 1995, 75-85 (Part 2).

[6] G. Strang, Wavelets and Dilation Equations: A Brief Introduction, *SIAM Review*, (31)4, December 1989, 614-627.