A Multimedia Framework for Effective Language Training

To appear, Computers & Graphics, Elsevier, 2007
ETH Zurich, Technical Report 570

Markus Gross, ETH Zurich, Christian Voegeli, Dybuster Inc.

Abstract

We present a novel framework for the multimodal display of words using topological, appearance, and auditory representations. The methods are designed for effective language training and serve as a learning aid for individuals with dyslexia. Our topological code decomposes the word into its syllables and displays it graphically as a tree structure. The appearance code assigns color attributes and shape primitives to each letter and takes into account conditional symbol probabilities, code ambiguities, and phonologically confusable letter combinations. An additional auditory code assigns midi events to each symbol and thus generates a melody for each input string. The entire framework is based on information theory and utilizes a Markovian language model derived from linguistic analysis of language corpora for English, French, and German. For effective word repetition a selection controller adapts to the user's state and optimizes the learning process by minimizing error entropy. The performance of the method was evaluated in a large scale experimental study involving 80 dyslexic and non-dyslexic children. The results show significant improvements in writing skills in both groups after small amounts of daily training. Our approach combines findings from 3D computer graphics, visualization, linguistics, perception, psychology, and information theory.

Key words: Visualization, Information Theory, Visual Information, Coding, Entropy, Learning, Natural Language, Dyslexia

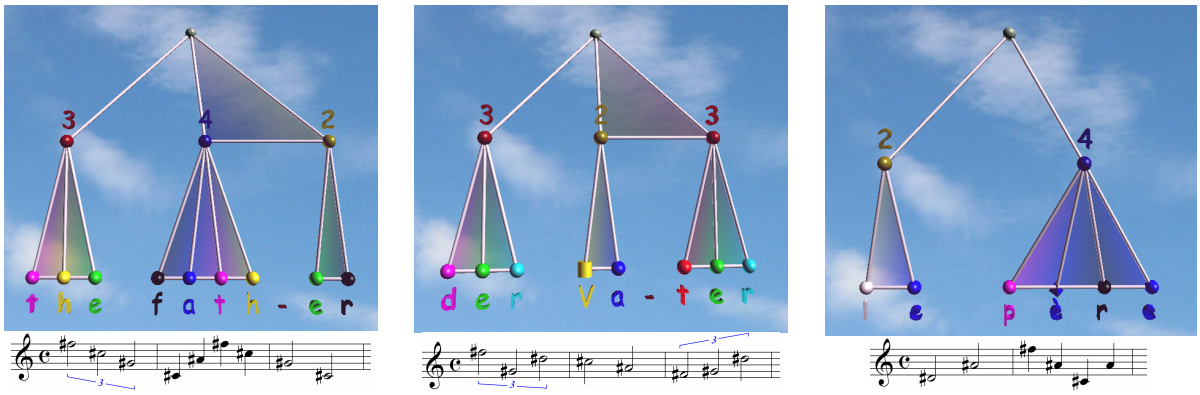
1. Introduction

Dyslexia is traditionally defined as the inability of otherwise intelligent individuals to acquire fluent reading and/or orthographically correct writing skills [1]. The socio-economical implications of dyslexia are significant and often devastating for the individual, who, in many cases, dramatically underperforms in school and profession. Dyslexia occurs predominantly in Western world languages, including English, French, German, or Spanish [1]. It is estimated that about 5-7% of the Western world population suffers from minor or major forms of dyslexia [2].

Dyslexia appears in various forms, such as *deep* or *surface*, and *developmental* or *acquired* dyslexia and at different levels of severity and strength. There are multiple causes for dyslexia, which, as of today, are not fully researched yet. Most often, dyslexia develops in early childhood and adolescence [3]. The irregularities in cerebral information processing underlying dyslexia are not fully understood yet and still subject of intensive research in psychology, medicine, neuroscience, linguistics, and other disciplines. A full overview of the exhaustive scientific literature on this subject is beyond the

[1] Supplemental materials submitted to Elsevier contain two videos demonstrating the system in action and a video report from a German TV station about the system and the user study.

[2] M. Gross (corresponding author) is with the Computer Graphics Laboratory, ETH Zurich, Switzerland. This work has been performed while the co-author worked as a junior researcher at this laboratory. Email: grossm@inf.ethz.ch, cvoegeli@dybuster.com



(a) Case I (b) Case II (c) Case III

Fig. 1. The string "the father" is displayed using a topological, color, shape, and auditory representation. The recoding retains information in that its overall entropy rate matches the entropy rate of the input symbols. The color code is language specific and the result of an optimization. We depict the codes for English, French, and German.

scope of this paper. We will confine ourselves to a summary of the most important findings relevant to our own work.

Researchers have proposed various models for the acquisition of human reading and writing skills. It is widely believed that orthographically correct writing is acquired over three phases: a *visual* phase, a *phonetic* phase, and a final *semantic* phase [2]. A traditional theory for reading is the *dual route model* distinguishing between a *phonological* and a *lexical* route [4].

More recent theories attribute dyslexia to a neurological disorder with a genetic origin and one school of thought explains dyslexia as a consequence of deficits in the phonological processing of the brain [5]. Scientists [6] also observed correlations between the occurrence of dyslexia and low level transient information processing in the human brain. Another school of thought [7] suggests that dyslexia is caused by specific weaknesses in visual and attentional processes. In particular, there is evidence for deficits in the transient, visual-temporal information processing - as opposed to visual-spatial perception, which is usually well developed in people with dyslexia. Such transient visual activity can be affected by the use of color [8]. A further theory [9] suggests that normal development and disorders of speech perception are both linked to temporospectral auditory processing speed. This key observation has been confirmed by various other authors [10].

Lately neurobiological evidence for reading and writing disorders has been given [11], [12], [13]. These researchers found abnormalities in the structure of the temporal cortex of dyslexic children using diffusion tensor imaging. Overall, these recent neurobiological studies seem to confirm the hypothesis that the difficulty in precise auditory timing has a link to language acquisition and comprehension [14]. It has also been hypothesized [15] that musical training may be able to remedy such timing difficulties. Our method compiles all these experimental findings into a novel, multimodal word recoding scheme.

1.1. *Therapy and Training Programs*

Numerous therapies of dyslexia have been proposed and applied so far. For instance, a French team showed [16] that a focused and abstract audio-visual training can lead to plastic neural changes in the cortex and thus improve cerebral language processing. Another very successful and scientifically well-founded therapy [9] utilizes the results from above. They developed a series of neuroplasiticity-based training programs that are designed to improve fundamental aspects of oral and written language comprehension and fluency. A further example for dyslexia treatment is LEXY [17], a Dutch therapy. This concept focuses on lexico-phonological deficits and employs the syllabic structure of words as its central element. Besides such scientifically well-founded approaches, there is a wealth of more or less heuristic therapies and learning aids. A comprehensive survey is given in [18], [19]. Various commercial multimedia e-learning systems [20] offer computer-based exercises to link words to their se-

mantics and to pictorial information. Strydom and du Plessis [21], for example, present a compendium of cognitive exercises aimed at the development of reading, writing, and other skills partly using color to support learning. Davis and Brown [22] depict words as 3-dimensional associations or as scenes sculpted by the user with play dough. While this method associates each word with a spatial representation, it is very cumbersome and of limited success. Overall and while significant progress has been made [9], there is no single commonly agreed-upon therapy for dyslexia to the present day.

1.2. *Our Approach*

The approach presented in this paper is fundamentally different from earlier ones in that it combines concepts from visualization and perception (in parts also used by e.g. [21] and [22]) with enhanced concepts from 3D graphics, statistical modeling of language, and information theory to design an advanced learning and language acquisition aid for individuals with dyslexia. The heart of our method is an abstract, graphical recoding of the input word. The code consists of a spatio-topological code, an appearance code (color, shape), and an auditory code, as displayed in Fig. 1. This coding transforms the input into a multimodal representation that supports phoneme-grapheme associations. It thus bypasses the distorted cognitive cues of people with dyslexia and builds alternative cerebral retrieval structures. A central feature of our recoding is its ability to measure and retain information through *entropy*. We take into account language statistics, code ambiguities, dyslexic letter pairs, and entropy maximization.

Our main contribution to computer graphics and visualization is a class of codes, which, to our knowledge, for the first time quantifies the visual recoding of information, *the* central processing stage in the visual display of data. The presented mathematical framework is versatile and can be generalized to other problems in graphics and visualization.

Besides dyslexia research there is a number of related scientific fields relevant to our work. The state-of-the art in those areas cannot be covered in this paper. Instead, we will refer to textbook literature. Our method relies heavily on statistical analysis of language such as being well-studied in linguistics [23]. The underlying mathematical models and coding algorithms draw upon Shannonian information theory and we refer to the textbook of [24]. Our core visual recoding algorithms can be considered as metaphors for the visual display of abstract information and thus relate to information visualization. The textbooks by Ware [25] and Tufte [26] are both very good examples in this area. The design of the presented method was influenced by our experience in perceptual aspects of data visualization [27]. Finally, the technical essence of the paper encompasses the quantification of multimodal information and thus stands in line with some significant recent research regarding visual importance [28], saliency [29], and others.

The paper is organized as follows: Section 2 summarizes the most important results of our statistical language analysis. We also give a short overview of entropy computation. Section 3 focuses on the design of the topological, shape, color, and auditory codes. Section 4 is devoted to the computational algorithms underlying the color and shape codes. Section 5 elaborates on the word selection controller, the core control unit of our learning system. The results of the experimental studies and evaluations of our method are detailed in Section 6.

2. Language Statistics and Entropy

Our paradigms for recoding are based on fundamental statistical properties of languages. These statistics were computed by linguistic analysis using commonly accepted language corpora, such as the British National Corpus for English (BNC) [30] and the European Corpus Initiative Multilingual Corpus [31] for German (ECIGer) and French (ECIFr). The BNC, for instance, comprises statistically relevant text fragments of contemporary language containing more than 95 Mio. words and 334,914 different English words. We will confine our summary of this analysis to some fundamentals as well as to the main findings that are relevant to understand and replicate the design of our codes. For the reader's convenience, we will focus on English.

2.1. *Word Frequency*

One of the fundamental laws in language statistics describes the frequency distribution of words. Zipf observed in 1949 [32] that the relationship between the rank of a word and its frequency follows a hyperbolic characteristic. Zipf's law states that the relationship between the rank r of a word and its frequency remains constant. Table 1 presents some

Table 1
Examples of word frequencies in % for English (BNC), French (ECIFre), and German (ECIGer) as found in our analysis.

Rank	English		French		German	
	Word	Freq.	Word	Freq.	Word	Freq.
1	the	5.68	de	5.51	der	3.31
2	of	3.17	la	2.81	die	2.96
3	to	2.69	l´	2.19	und	2.29
4	and	2.63	à	2.03	in	1.73
5	a	2.14	le	2.01	den	1.17
10	for	0.87	en	1.32	im	0.80
100	after	0.099	francs	0.086	gibt	0.083
1000	win	0.010	principal	0.010	Nähe	0.0091

of the most frequent words in English, French, and German.

To design the word selection controller (section 5.4) and training database for our learning system (section 5.2) it is necessary to rank the most frequent words. In practice, of course, we have to limit the database to about 70%-90% of the most frequent words of a language. It turns out, for instance, that the 8000 most frequently used words in English and French cover 90% of the corpora while 34000 words of German are needed to achieve 90% coverage. This is due to the typical compound noun constructions of German language. *"Kinder-garten"* is a simple example.

2.2. *Word Length Distribution*

A further important statistics of a natural language is its distribution $P_w(J)$ of the word length J. For our purposes, specific word length distributions are required, both as a function of the word's number of letters and as a function of the number of its syllables. It has been noted by several authors [33] that Poisson distributions and binomial distributions are well suited to model this language property.

A refined fit can be obtained by log-Weibull distributions. For practical use, we limit the number of letters of a word to a maximum $\max(J) = 25$. The normalized log-Weibull distribution $P_{w_{\text{weib}}}$ yields as

$$P_{w_{\text{weib}}}(J) = \frac{e^{(a-J)/b - e^{(a-J)/b}}}{b \sum_{i=1}^{\max(J)} P_{\text{weib}}(i)}, \; J \in \{1, ..., \max(J)\}. \quad (1)$$

a and b are language specific parameters. Fig. 2 depicts the results of our analysis, where we compare

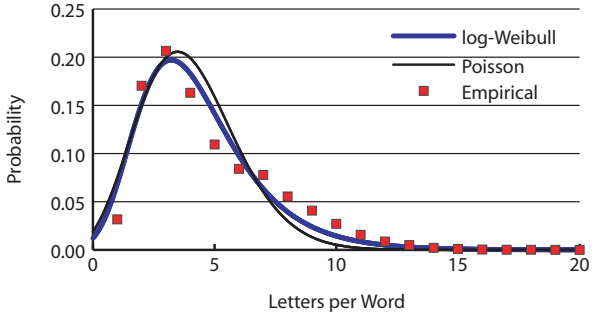


Fig. 2. Word length distribution of the BNC. We compare the empirical data with the Poisson distribution ($\mu = 3.99$, mean 4.065) as well as the log-Weibull distribution ($a = 3.22$ and $b = 1.89$).

Poisson and log-Weibull statistics for word length distributions fitted to the BNC data. Model and data match very well.

It is very instructive to compare the average word lengths for different languages. Our analysis revealed that the average word length for English is 4.73 letters per word, compared to 6.17 for German and 4.88 for French.

2.3. *Syllable per Word Distribution*

Another important linguistic statistics constitutes the distribution $P_y(K)$ of the number K of syllables y per word. It is important to note that not all researchers count syllables the same way. For our purpose we utilized an implementation of Knuth's hyphenation algorithm [34] which is also employed to construct the syllable tree in Section 3.2. For our analysis we limit the maximum number of syllables per word to $\max(K) = 10$.

2.4. *Syllable Length Distribution*

A further important input to our recoding method is the distribution $P_s(L)$ of syllable lengths L. While this statistics can be modeled using Poisson distributions as well, we found that the Conway-Maxwell distribution provides a much better fit. For practical utility we limit the maximum number of letters per syllable to $\max(L) = 15$ and yield

$$P_s(L) = \frac{a^L}{L!^b} C_1, \quad L \in \{1, ..., \max(L)\}, \quad (2)$$

with a, b being language specific parameters and C_1 being a normalizing term.

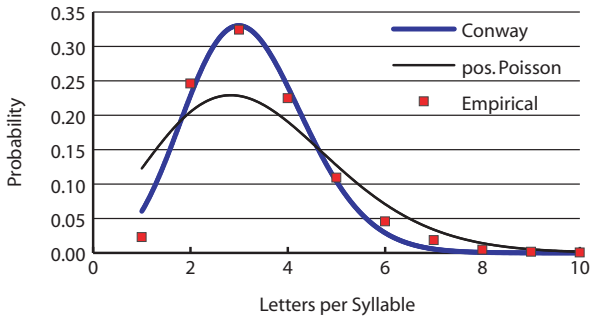


Fig. 3. Syllable length distribution of the BNC. We compare again the empirical data to fits of a Poisson distribution with $\mu = 3.99$ and a Conway-Maxwell distribution with $a = 19.53$, $b = 2.37$ and $C_1 = 0.0031$.

Table 2
Probabilities of the 3 types of symbols employed by the topological encoder of Section 3.2 and their entropy.

Statistics	**English**	**German**	**French**
$\mathbf{P}(\circ)$	0.2115	0.1619	0.2046
$\mathbf{P}(-)$	0.0812	0.1661	0.1142
$\mathbf{P}(*)$	0.7073	0.6770	0.6812
$H(Y)$	1.121	1.231	1.203

Fig. 3 compares the model to the data of the BNC and illustrates the quality of this distribution. The average length of a syllable in English is computed as 3.40, while we obtain 3.08 for German and 3.13 for French.

2.5. *Special Symbols*

As we will explain in more detail in Section 3.2 our topological code distinguishes between the following three different types of symbols:

° : symbol for letter marking the end of a word
- : symbol for letter marking the end of a syllable, but not the end of a word
* : symbol for a regular letter

The computed probabilities for these symbols are listed in Table 2.

2.6. *Symbol Probability and Markov Models*

A key ingredient for the design of any text coding method is the symbol statistics [35], where the sequence of symbols is represented by a random variable X. Depending on the order of the underlying Markov model we have to compute more or less complex conditional symbol probabilities.

Table 3
Probabilities of capital letters and umlauts/accents for English, German, and French, as well as the entropy of their distribution.

Statistics	**English**	**German**	**French**
capital letters	2.949%	6.675%	2.169%
umlauts	0%	1.600%	3.030%
small umlauts	0%	1.560%	3.028%
$H(S)$	0.1918 bits	0.4681 bits	0.3456 bits

Let $\mathbf{A}$ be an alphabet of size $|\mathbf{A}|$ with $|\mathbf{A}|$ being the number of symbols (letters) $x_i \in \mathbf{A}$. The Markov-0 model is fully described by the probabilities of occurrence $P(x_i)$ of symbol x_i. In this model, a string $\mathbf{s} = (a_1, ..., a_J)$ of symbols a_j and length J is considered as a random sequence and the occurrence of a symbol $a_j = x_i$ is *statistically independent* of preceding symbols of that string.

A somewhat more elaborate model is Markov-1, where the occurrence of a symbol a_j in the string $\mathbf{s}$ is statistically dependent on the preceding symbol of the sequence. This dependency is expressed by the conditional probability $P(a_{j+1} = x_i | a_j = x_k)$. Such conditional pairs are called digrams. Higher order Markov models utilize trigrams, quadgrams and so forth [35].

For the computation of conditional probabilities all special characters were omitted except the space character to separate words. We also distinguish capitalized letters. For German and French, umlauts, such as {ö, ü, ä, è, à...} were taken into account as well. Our analysis showed that the most frequent digram in the BNC is actually *"e_"* which symbolizes an *"e"* at the end of a word. The second most frequent digram is *"_t"*, or a *"t"* at the beginning of a word.

It is clear that the symbol probabilities differ significantly between languages. For instance when considering the space character "_" as part of the alphabet, the probability of *"e/E"* is 12.4% for English, 13.6% for German and 11.8% for French including all accents.

The shape code (section 3.3), which is part of the appearance encoder, assigns different shape primitives to capitalized, umlauted, or accented letters. The probabilities for such letters are listed in Table 3.

The results confirm that German contains more capital letters (all nouns) while French comprises most accents. The number of capitalized umlauts and accents is negligible for the design of the code.

2.7. *Information and Entropy*

The random process described by the variable X is considered as an information source. To measure this information, we employ the concept of entropy H [36]. We assume that the random process X is stationary and memoryless allowing us to compute *entropy rates*. For convenience of notation, we will utilize the term entropy in subsequent analysis. H measures the entropy rate of the associated Markov source, that is, the average number of bits needed to encode a single symbol x_i and can be computed as

$$H_{M_0}(X) = -\sum_{i=1, P_X(x_i)\neq 0}^{|\mathbf{A}|} P_X(x_i)\log(P_X(x_i)) \quad (3)$$

measured in bits/symbol.

In our analysis, we obtain an entropy for English of about 4.18 bits/symbol if we do not consider the space character x_0, of 4.11 bits/symbol if we do, and of 4.29 bits/symbol if we additionally discern small and capital letters. For German we get 4.50 bits/symbol including x_0, umlauts and capital letters and 4.12 bits/symbol without including them. French results in 4.29 and 4.03 bits/symbol respectively. To encode an English text string $\mathbf{s}$ of length J using a Markov-0 model we thus need on average $JH(X)$ bits. In a similar fashion entropy can be computed in a Markov-1 model.

Our analysis of the BNC yields a Markov-1 symbol entropy of $H(X) = 3.48$ bits for English, $H(X) = 3.51$ for German, and $H(X) = 3.40$ for French, all including capital letters. As expected, the rates for Markov-1 models are lower, since such models comprise less uncertainty due to the improved conditional symbol statistics. The string entropy for a Markov-1 model can be computed accordingly. For more fundamentals on information theory and important theorems related to entropy we refer the reader to textbooks, such as [24], [35].

From now on, we will refer to $H(X)$ as the *symbol entropy*.

3. The Design of Visual and Auditory Codes

In this section, we will describe the design and computation of the individual codes we utilize for the recoding of a given input string $\mathbf{s}$. Central to our design is to retain entropy during the recoding. For this purpose, we quantify the visual and auditory information by estimating their entropies.

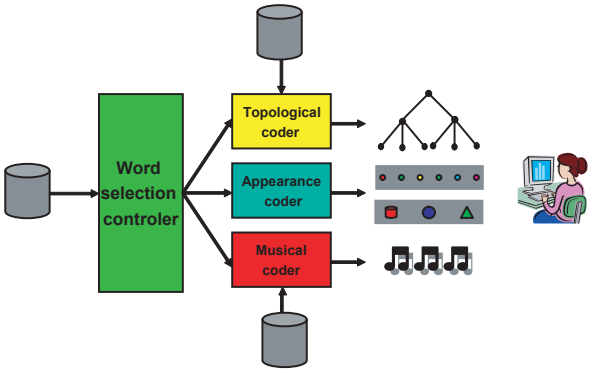


Fig. 4. Conceptual components of our method and framework.

3.1. *Overview of the Recoding Method*

Fig. 4 shows the general concept. An input string $\mathbf{s}$ (e.g. a word) is taken from a training dictionary by a word selection controller. The controller adapts to the user's learning state and minimizes error entropy (section 5). The recoding of $\mathbf{s}$ is accomplished by three different encoders: the topological encoder (section 3.2), the appearance encoder (section 3.3), and the auditory encoder (section 3.4), which all together create a multimodal representation of $\mathbf{s}$.

Fig. 5 depicts a result. In a first step we parse the string recursively and decompose it into a syllable tree. This tree generates a topological representation of the syllable structure of the string. We refer to it as the *topological code* of $\mathbf{s}$. Wherever a blank occurs the string is separated into a set of words. Next, each word is decomposed into a set of syllables. To accomplish this, we use a hyphenation algorithm similar to the one suggested by Knuth [34], and insert hyphens into the original string. Each syllable is regarded as a set of letters. From this decomposition, a tree is constructed, where the string $\mathbf{s}$ is associated with the root. The internal nodes represent all syllables of $\mathbf{s}$. If two syllables belong to the same word, their nodes are joined by a horizontal bar (see Fig. 5). Also, we assign a number to each internal node depicting the number of letters of the syllable. In addition, a unique color is used to encode the syllable length. The letters are represented by the leave nodes. This representation is a fully interactive and animated 3-D structure. We will explain interaction in more detail in Section 5.

The leaf nodes are represented by the *appearance codes* of $\mathbf{s}$. To this end each letter $x_i \in \mathbf{A}$, is assigned a color value $c_k \in \mathbf{C}$ from the color alphabet $\mathbf{C}$. The map $c(x_i)$ projecting each letter to color values will

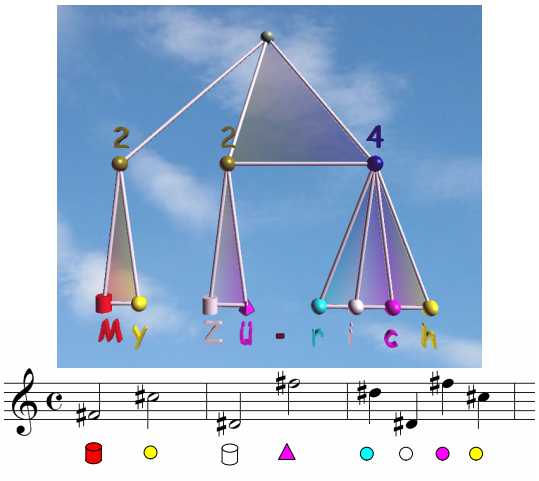


Fig. 5. Representation of the string "My Zürich" including the color code, the shape code, the topological code and the auditory code.

be elaborated below. Similarly, we assign a shape primitive s_n from a shape alphabet $\mathbf{S}$ to each letter x_i using a map $s(x_i)$. While our implementation and analysis confines appearance to shape and color, other features, such as transparency, reflection, surface microstructure, texture, or changes thereof over time can be included easily. The current setting utilizes the shape code to distinguish between regular letters (sphere), capital letters (cylinder), and umlauts (tetra).

Finally, we assign an *auditory code* to the representation. In the current approach, the auditory code translates the visual representation, i.e. its topological and appearance code, into a set of midi events. The musical attributes we use include pitch for color and instrument for shape, as well as duration and rhythm for syllable lengths. Thus, the auditory code eventually creates a melody for $\mathbf{s}$, which is played by the computer's midi synthesizer to reinforce the visual codes. Fig. 5 displays the score of *"My Zürich"* as assigned by the auditory encoder. Any aspects of a musical or auditory event that enables the human auditory system to distinguish two, otherwise similar musical events from each other could be included. Examples are volume, scale, reverb, style elements, chords, harmony, and the like.

While the topological and shape codes are determined by $\mathbf{s}$ by construction, the color code leaves some degrees of freedom such as the number of colors and the actual color values. We will first present the codes in more detail, show how they are linked

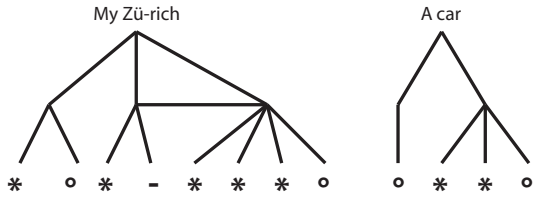


Fig. 6. topological code for the strings "My Zürich" and "A car".

to each other by entropy and then elaborate on the optimal choice of the free parameters of the color code.

3.2. *Topological Coding*

Our goal is to measure the entropy of the graph. In graph theory, a well-known result [37] states that a general tree $T(V, E)$ with V nodes and E edges can be encoded in $2(V + 1)$ bits, where V is the number of nodes. For our purposes, we designed an efficient code to approximate the *tree entropy* taking advantage of the inherent tree structure. For instance, our syllable tree always features three levels. The central idea is to construct a *ternary code* with the associated entropy $H(Y)$. As explained in section 2.5 we utilize a distinct symbol "°" for letters at the end of a word, "-" for letters at the end of a syllable which do not mark the end of a word, and "*" for all other letters.

The resulting alphabet of the code yields as $Y = \{\circ, -, *\}$. Fig. 6 gives an example. As can be seen, the code string " * ° * - * * * ° " e.g. encodes the topological information of the tree including the horizontal bar.

The associated symbol probabilities are given in Table 2. Y is a Markovian random variable and its entropy can be computed from the symbol probabilities of the ternary alphabet. Since Y unambiguously encodes T, the entropy of T is bound by the entropy of Y. We have not proven the optimality of this code yet. However, since syllable ends and word ends have to be distinguishable, other Markov-0 codes will not be much more efficient. Hence, $H(Y)$ is a conservative estimate for the entropy of T. Our analysis gives $H(Y) = 1.12$ for English, $H(Y) = 1.23$ for German, and $H(Y) = 1.20$ for French. We multiply $H(Y)$ with the average word length as given in Section 2.2 and thus obtain $1.12 \cdot 4.73 = 5.30$ bits as a tree entropy for English.

To formalize the information of the numbers associated with each internal node (see Fig. 5), we

introduce the random variable N and the *number entropy* $H(N)$. Since the tree can be decoded from the ternary code, the basic theorems of information theory state that the conditional information

$$H(N|Y) = 0. \tag{4}$$

The Markovian source N generates a number per syllable representing its length. Knowing the syllable length distribution $P_s(L)$ from Section 2.4 we compute $H(N) = 0.692$ bits per symbol for English, 0.605 b/s for German and 0.715 b/s for French.

3.3. *Appearance Coding*

The appearance coder includes a color code to represent a letter and a shape code to distinguish between regular letters, capital letters and umlauts. The mathematical details of the computation of the color code are deferred to Section 4.

Let $c(x_i)$ be the color code that maps each symbol x_i onto a color c_k, where c_k belongs to the set $\mathbf{C} = \{c_1, ..., c_{|\mathbf{C}|}\}$. We compute the probability of occurrence of a color c_k in $\mathbf{A}$ by summing up over the probabilities of all symbols x_i mapped to c_k

$$P(c_k) = \sum_{i=1}^{N} P(x_i|c(x_i) = c_k). \tag{5}$$

Again, we assume C being a random Markovian process and compute the *color entropy* $H(C)$ as

$$H(C) = -\sum_{k=1}^{|\mathbf{C}|} P(c_k)\log(P(c_k)), \tag{6}$$

where $|\mathbf{C}|$ stands for the number of colors.

Note that maximizing $H(C)$ implies

$$P(c_k) = \frac{1}{|\mathbf{C}|} \Leftrightarrow H(C) = \log(|\mathbf{C}|), \tag{7}$$

i.e. a *uniform* probability distribution of $P(c_k)$ for the occurrence of a color. This property will be exploited in Section 4 to compute the color code.

In a similar fashion we can compute the *shape entropy*. Fig. 7 depicts the three different shape primitives we apply. A sphere represents a regular, non-capitalized letter, a cylinder stands for a capitalized letter, and the tetrahedron encodes a special letter, in particular an umlaut in German or an accent in French. The rationale for choosing spheres, cylinders, and cones is their symmetry and simplicity resulting in low shape entropy.

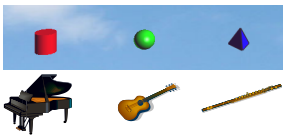


Fig. 7. Three different shapes and instruments used to encode special letters.

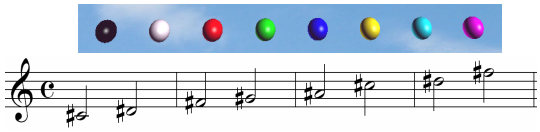


Fig. 8. Auditory code using pitch to reinforce the color code.

Let $\mathbf{S} = \{s_1 = \text{sphere}, s_2 = \text{cylinder}, s_3 = \text{tetra}\}$ be the set of shape primitives and let S be the Markovian random process describing the occurrence of such shapes. Hence the shape entropy $H(S)$ is given by the shape probabilities $P(s_i)$ listed in Table 3. Our analysis resulted in $H(S) = 0.192$ for English, $H(S) = 0.468$ for German and $H(S) = 0.346$ for French. These results confirm our intuition, that special letters occur on average more frequently in French and German.

3.4. *Auditory Encoding*

The auditory coder first assigns a musical note with a distinct pitch to each color generated by the color coder. Fig. 8 shows the current assignment. We experimented with different pitches, keys, and scales, including major, minor, chromatic, Jazz, Klatzmer, and others, but found that a pentatonic scale is the most simple one. It is known that random combinations of notes of a pentatonic scale lead to aesthetically pleasing melodies, which leads to the creation of the most pleasing word melodies.

Furthermore, the coder assigns note length to syllable length. For instance, a two-letter syllable is mapped to half notes, a three-letter syllable to a triplet, a four-letter syllable to a quarter, and so forth. This scheme replays syllables with more letters faster and enables the user's auditory system to distinguish between longer and shorter syllables. Fig. 9 presents the assignment.

Finally, we represent the shape code by means of different musical instruments. Table 4 summarizes how the color, topological, and shape codes are translated to midi events.

In order to formalize the entropy of the auditory code we introduce the Markovian random variable

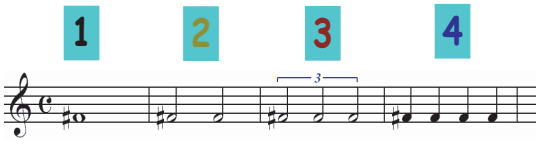


Fig. 9. Auditory code using note length to encode syllable length and to reinforce the topological code.

Table 4
Midi events assigned to the color, topological and shape codes.

Code	Event
Color	pitch(pentatonic)
Shape (small cap)	guitar
Shape (large cap)	piano
Shape (umlaut)	flute
Tree	note length

M denoting the random process which creates a triplet {instrument, pitch, length} for each letter. By construction the following relation holds for the conditional entropy of the auditory code:

$$H(M|YCS) = 0. \tag{8}$$

This equation can be summarized as follows: while the auditory code reinforces the visual codes through an additional perceptual cue, it does not provide additional information in a Shannonian, information theoretical sense. Note also that the above analysis is confined to information related to the random process that creates the string $\mathbf{s}$. In practice, the generated midi sounds contain other information as well, such as touch etc., but such information is irrelevant for our purposes.

The assignments of musical attributes to visual elements were done empirically. Word rhythm, for instance, is reflected by a words syllabic structure, whereas color corresponds to different spectral frequencies and is thus mapped to pitch. As for the instruments, we picked the ones that pleased a majority of individuals testing our software.

3.5. *Retaining Letter Entropy*

One of the free parameters of the color code is the total number of colors $|\mathbf{C}|$. We need to compute this number in order to design the color code in Section 4. Central to this computation is to *retain the information* of the word. This means that the total information represented by the topological, appearance, and auditory encodings should be no less than the Markov information of the original string $\mathbf{s}$ and should allow to unambigously reconstruct $\mathbf{s}$. In terms of entropy, this requirement translates into the condition

$$H(X|YNMCS) = 0, \tag{9}$$

which implies by the laws of information theory [36] that

$$H(YNMCS) \geq H(X). \tag{10}$$

N is determined by Y in the same way as M is defined by Y, C and S, see (8). Hence we get $H(YNCMS) = H(YCS)$. For our practical computation of $|\mathbf{C}|$ we reformulate (10) as

$$H(YCS) = H(YNMCS) = \lambda H(X), \quad \lambda \geq 1. \tag{11}$$

This equation actually compares the information of a letter $H(X)$ with our recoding assuming Y, C, and S are statistically independent, which is not the case. The fundamental theorems, however, state that

$$H(YCS) \leq H(Y) + H(C) + H(S) \tag{12}$$

and make the right-hand side sum a conservative estimate of $H(YCS)$. While the sum contains additional redundancy,

$$H(Y) + H(C) + H(S) = \lambda H(X) \tag{13}$$

is a good design choice.

The maximum of $H(C)$ is given by (7). Inserting it yields

$$\log(|\mathbf{C}|) = H(C) = \lambda H(X) - H(Y) - H(S) \\ \Rightarrow |\mathbf{C}| = 2^{\lambda H(X) - H(Y) - H(S)}. \tag{14}$$

The results of our analysis are summarized in Table 5. We present all involved entropies. Requiring $\lambda = 1$ provides a fractional result for the number of colors denoted by $|\mathbf{C}_1|$. In practice, we have to round up or down to the next integer number, choosing $|\mathbf{C}| = 7$ for German and French and $|\mathbf{C}| = 8$ for English. However, since our code introduces redundancy (12), we factor in some safety and set $|\mathbf{C}| = 8$ for all three languages.

It deserves discussion that this choice $|\mathbf{C}| = 8$ does *not* guarantee that our recoding is lossless, but it gives a lower bound for guaranteed information loss if we chose less than $|\mathbf{C}_1|$ colors. It is noteworthy that a precise computation of the joint entropy $H(YCS)$ could be accomplished using the chain rule for entropies,

$$H(YCS) = H(Y) + H(C|Y) + H(S|YC), \tag{15}$$

Table 5
Summary of the model related entropies for English, German, and French.

	English	German	French
$H(X)$	4.29	4.50	4.29
$H(Y)$	1.121	1.231	1.203
$H(S)$	0.192	0.468	0.346
$H(C)$ if $\lambda = 1$	2.976	2.802	2.741
$\lvert\mathbf{C}_1\rvert$	7.867	6.972	6.686
λ if $\lvert\mathbf{C}\rvert = 8$	1.006	1.044	1.060
$H(N)$	0.692	0.605	0.715

but the conditional entropies involved are very hard to compute. Again, the lack of information loss is a necessary condition for the existence of a unique code. We will get back to this issue in the following section.

4. Computation of Color Codes

To compute the color code we recall that $H(M|YCS) = 0$ so that the auditory code does not provide any additional information and that the topological and shape codes are given by $\mathbf{s}$. Hence we are left to optimize the color code $c(x_i)$. $c(x_i)$ defines the assignment of a color c_k to each symbol $x_i \in \mathbf{A}$. This code can be viewed as a discrete map. As we will explain in detail below, the code is the result of a discrete optimization procedure minimizing a multi-objective function E. This function takes into account the following terms:

4.1. *Uniform Color Distribution*

First, we want to assign the colors to individual symbols in such a way that the color entropy is maximized. As discussed in Section 3.3, this is achieved by a uniform distribution of $P(c_k)$. To this end, we define a so-called *color energy* E_C as part of the objective function

$$E_C = \omega_C \sum_{i=1}^{|\mathbf{C}|} \sum_{j=1}^{|\mathbf{C}|} |P(c_i) - P(c_j)|, \tag{16}$$

with ω_C being a weight. It is easy to see that this energy is minimized for $P(c_k) = 1/|\mathbf{C}|$. It turns out that for highly uneven symbol probabilities $P(x_i)$ too many symbols are potentially mapped to the same color. Therefore, it is useful to try to limit the number of symbols mapped to a color between $\lfloor |\mathbf{A}|/|\mathbf{C}| \rfloor$ and $\lfloor |\mathbf{A}|/|\mathbf{C}| \rfloor + 1$. This can be achieved

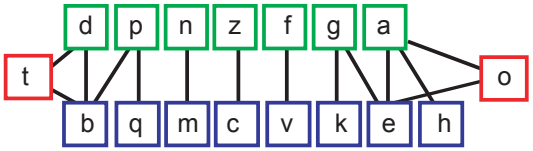


Fig. 10. Dyslexic pair constraints used to compute the color code.

by choosing the weight ω_C as the following soft constraint:

$$\omega_C = \begin{cases} 1, & c_{\min} \leq \sum_{k=1, c(x_k)=c_i}^{N} 1 \leq c_{\max}, \ \forall c_i \\ c_{\text{soft}}, & \text{else} \end{cases}, \tag{17}$$

where $c_{\min}$ is a lower bound for the number of symbols per color, $c_{\max}$ is an upper bound for number of symbols per colors and c_{soft} is a user penalty. It should be noted, however, that these constraints have to be set with care. In German and French, for example, the probability of *"e"* is more than 15%, if the space character is not considered as a part of the alphabet. In our simulations, we set $c_{\text{soft}} = 2$. This way the constraint affects the final solution by about 15% in German, such that assignments violating the lower and upper bounds are hardly possible.

4.2. *Dyslexic Pairs*

It is well-known in dyslexia research and therapy that phonetically similar symbols pose specific difficulties for people with dyslexia. Such "*dyslexic pairs*" include *"d-t"*, *"p-b"*, *"m-n"* and others, as well as silent consonants, such as *"h"*. Fig. 10 shows all the symbol pairs we consider. They comprise stop sounds, symmetric pairs, and triplets as well. This set is partly language specific and can be altered and extended.

For optimization, we first define a so-called *dyslexic map* $\text{dys}(x_i, x_j)$ for any symbol pair $(x_i, x_j) \in \mathbf{A}^2$. This map takes a value of 1 for dyslexic pairs, and 0 otherwise. Hence

$$\text{dys}(x_i, x_j) = \begin{cases} 1, & \text{if } (x_i, x_j) \text{ dyslexic pair} \\ 0, & \text{else} \end{cases}. \tag{18}$$

Our goal is to map such pairs onto appearance attributes with a large perceptual distance, denoted by the norm $d_P(\cdot, \cdot)$. We want to map, for instance, the pair *"t"* - *"d"* onto the colors red and green, respectively. To quantify the quality of our mapping, we define a *dyslexic energy* E_D with

$$E_D = \omega_D \sum_{k=1}^{N} \sum_{l=1}^{N} \frac{dys(x_k, x_l)}{1 + d_P(c(x_k) - c(x_l))} \tag{19}$$

While the above formulation sets a soft constraint on dyslexic pairs, d_P allows us to control its importance. We will address the perceptual color norm in Section 4.5.

4.3. *Frequent Letter Pairs*

In recent years, scientific evidence has hardened that people with dyslexia exhibit significant distortion in the perception of rapid temporal information [9]. Very often, they read letters in the wrong order, i.e. *"hopsital"* instead of *"hospital"*. To address this phenomenon, we try to avoid frequent symbol pairs, such as *"h-o"*, *"o-s"*, *"s-i"* being mapped to the same color. Of course, since $|\mathbf{C}| < N$, there is no way to eliminate color repetition entirely. Yet, our Markov-1 statistics gives us means to include frequent symbol pairs into our cost function.

We define a *Markov-1 energy* E_{M_1} that assigns frequent letter pairs to different colors. This energy is similar to E_D, but pairs are weighted with their joint probability. We set

$$E_{M_1} = \omega_{M_1} \sum_{k=1}^{N} \sum_{l=1, l\neq k}^{N} \frac{P(x_l|x_k)P(x_k)}{1 + d_P(c(x_k), c(x_l))}, \tag{20}$$

where $P(x_l|x_k)$ is the conditional probability that symbol x_l follows after symbol x_k (digram probability). Note that pairs of the same symbols (*"aa"*, *"bb"*,...) are mapped to the same color and are therefore omitted in (20). It has turned out that the above soft constraint formulation retains more flexibility for the optimization than setting hard constraints for E_D and E_{M_1}.

4.4. *Unique Coding*

We assume for convenience that the input string $\mathbf{s}$ consists of a single word, hence $\mathbf{s} = \mathbf{w}$. Ideally, we want to recode each word $\mathbf{w}$ from the dictionary $\mathbf{D}$ such that no two words are mapped to the same code. It is important to emphasize that the entropy computations of (11) – (14) state that such a code *can exist*, but not all possible codes are unique. Therefore, we incorporate a uniqueness constraint into the optimization. In addition, some words are more difficult to learn than others. It is important to ensure that difficult words in $\mathbf{D}$ are assigned to unique code words, as opposed to simple ones. Likewise, more frequent words should be mapped to unique code words for better distinction. For this purpose, we define a *coding energy* E_U that strives to minimize code ambiguity while putting weight onto frequent and difficult words.

Let $P(\mathbf{w})$ be the word probability as given in Table 1 and let $\mathrm{diff}(\mathbf{w})$ be a function returning the difficulty level of a word. We first define a word weight function $W(\mathbf{w})$ as

$$W(\mathbf{w}) = P(\mathbf{w})\mathrm{diff}(\mathbf{w}) = P(\mathbf{w})\mathrm{dys}(\mathbf{w}) \cdot |\mathbf{w}|H(X). \tag{21}$$

$\mathrm{diff}(\mathbf{w})$ is basically the product of the length of the word $\mathbf{w}$, the symbol entropy, and the number of dyslexic pairs $dys(\mathbf{w})$ in $\mathbf{w}$. Using this definition, the coding energy yields to

$$E_U = \omega_U \sum_{\mathbf{w}\in\mathbf{D}} \tilde{W}(\mathbf{w}) \tag{22}$$

where

$$\tilde{W}(\mathbf{w}) = \begin{cases} 0, & \text{if } \mathbf{w} \text{ is uniquely coded} \\ W(\mathbf{w}), & \text{else} \end{cases} \tag{23}$$

We will utilize $\mathrm{diff}(\mathbf{w})$ again for the design of the word selection controller in Section 5.4.

4.5. *Color Attributes*

E_D and E_{M_1} include the evaluation of a color distance. Perceptual color spaces have been researched extensively in color science [38] and literature provides a variety of perceptually uniform color spaces, such as $L^*u^*v^*$, $L^*a^*b^*$, YMS or $l\alpha\beta$ [27]. These color spaces perform a nonlinear distortion of R, G, B or X, Y, Z in such a way that distances become nearly perceptually uniform and measure color distances using Euclidean norms.

Let c_k and c_l be two color attributes with coordinates (c_{kR}, c_{kG}, c_{kB}) and (c_{lR}, c_{lG}, c_{lB}) in some color space respectively. We define the distance $d_P(c_k, c_l)$ simply as the Euclidean distance between the coordinates of two color attributes, with

$$\begin{aligned} d_P(c_k, c_l) &= ||c_k - c_l|| \\ &= \sqrt{(c_{kR}-c_{lR})^2+(c_{kG}-c_{lG})^2+(c_{kB}-c_{lB})^2}. \end{aligned} \tag{24}$$

Given the total number of colors $|\mathbf{C}|$ the computation of the coordinates in color space involves the following optimization.

$$\max(\min(d_P(c_k, c_l))), \quad \forall k, l. \tag{25}$$

We experimented with different color spaces and maximized the above expression using simulated annealing. The visual results for the perceptually uniform spaces, however, are not substantially superior to the trivial solution in the R, G, B space. For $|\mathbf{C}| = 8$ we obtain slight variations of the corners of the RGB-cube. Similar computations were performed by [39].

4.6. *Minimizing the Cost Function E*

Minimizing E is a multi-objective discrete optimization problem, or a so-called Pareto problem [40]. Since the individual terms of E cannot be compared to each other, multiple "optimal" solutions exist in the search space. The set of all solutions that are not dominated by others are called the *Pareto front*. A general solution strategy for such problems is to aggregate the objectives into a weighted sum and to minimize it. Applied to our problem the overall cost $E(c(x))$ of a given color assignment $c(x)$ is computed by

$$E(c(x)) = \omega_C E_C + \omega_D E_D + \omega_{M_1} E_{M_1} + \omega_U E_U \quad (26)$$

with ω_D, ω_C, ω_U, ω_{M_1} being the weights.

Minimizing the cost function is highly nontrivial, since the search space encompasses $|\mathbf{C}|^N$ different mappings for $c(x)$. We designed various randomized assignment and search algorithms that run in combination to guarantee robust Pareto solutions. In order to compare the quality of the solutions on the Pareto front, we ran additional optimizations with only one of the four objectives each. The discrete randomized search is a two-pass method, where in a first step an initial symbol to color assignment $c(x)$ is computed. This initial guess is improved subsequently by randomized optimization.

In order to avoid the search getting stuck in local minima we ran the optimization with different strategies for the initial assignments. The idea is to distribute these initial values across the search space. We developed various initial assignment algorithms whose discussion is beyond the scope of this paper.

After the initial assignment a randomized discrete search algorithm computes permutations of $c(x)$ while minimizing the objective function. We designed four different methods which distinguish in the size of the neighborhood they search per iteration. All algorithms share the following sequence of operations: i) select a color, ii) select a letter, iii) assign color to letter. We combine all these algorithms to achieve best results. In practice, we compute an initial assignment and run two of the random search methods alternately until the solution is stable.

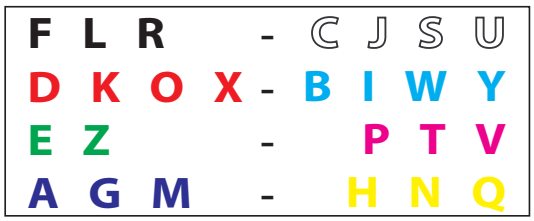


Fig. 11. Optimized color code for English.

It is noteworthy that efficient updates and evaluations of the objective function require efficient data structures. In particular, E_U poses great challenges, because it involves a check for unique coding. Naively, such checks require to compare all words in $\mathbf{D}$ and are hence $O(|\mathbf{D}|^2)$. By definition, however, the topological code of each word is determined by its syllabic structure and hence does not change during optimization. Thus, it is sufficient to compare only words with the same topological code.

For example, the 8000 word English dictionary covering 90% of the corpus comprises 244 different topological codes so that on average 32.8 words are mapped to the same syllable tree. Fig. 11 displays the computed color code for English. A more detailed discussion of the results can be found in Section 6.

5. Interactive Learning and Word Selection

In this section we will explain how the described recoding is utilized in an interactive multimodal language training system for individuals with dyslexia. After a summary description of the setup we will focus on the word selection controller which adapts to the user's actual state and guarantees optimal learning rates.

5.1. *System Concept and Interaction*

Our interactive language training system (*"Dybuster"* [41]) is a Windows based software that runs on conventional PCs with 3D graphics acceleration. While the system is designed for people with dyslexia, our user study proves that it can also be employed for effective training of non-dyslexic users. The software consists of three simple games: a color game, a graph game, and a word learn game.

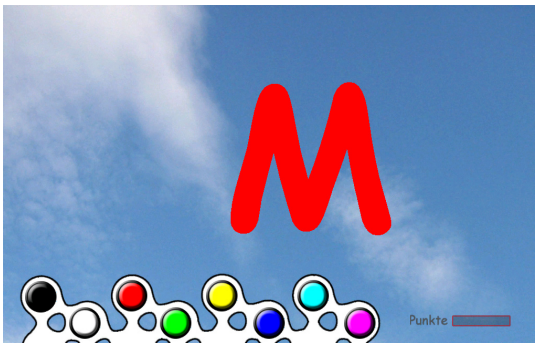


Fig. 12. Screen shot of the color game. The color buttons in the lower part of the screen are to prompt the color of the displayed symbol.

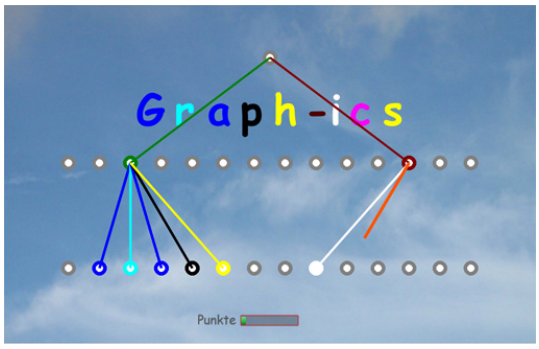


Fig. 13. Screen shot of the graph game. The array of nodes allows users to construct the topological encoding of the prompted word.

The purpose of the *color game* is to learn the color and auditory codes. To this end the system initially selects random symbols from $\mathbf{A}$ and presents them to the user who has to confirm the right color by mouse click. A screen shot of this game is shown in Fig. 12. As the user progresses, the color saturation of the presented symbol fades to white requiring the user to memorize and recall the color. A midi event playing the auditory code confirms a correct click. A score counter provides feedback on the actual learning state. During learning, the system computes a color confusion matrix and selects symbols based on prior error probabilities.

Likewise, the user learns the concept of topological encoding using a *graph game*, as shown in Fig. 13. In this game, the user must draw the correct tree of a given word by clicking onto arrays of nodes and by drawing a rubber band. Acoustic signals and musical events confirm correct clicks. The words are taken from our database $\mathbf{D}$.

The most important system component is the *word learn game*. Here, the system displays the spatial and color codes of a word selected from $\mathbf{D}$. The system also replays the auditory code (word melody) as well as a prerecorded wave-file of the pronunciation. The user must type in the word through the keyboard. Upon each keystroke, the auditory code of the typed symbol is played. A special acoustic signal indicates a spelling error. A score counter provides feedback on learning performance. Orthographic errors are tabulated and utilized to compute word error probability and other performance measures. The details of how to build $\mathbf{D}$ and how to select individual words will be given below.

The system also offers an input mode, allowing the user to feed new words into the database. A very large dictionary can optionally assist her to automatically complete words and to check for correct spelling. In addition, the user can choose an automatic hyphenation algorithm or alternatively hyphenate the words manually by adding hyphen symbols. Optionally, articles or longer strings with multiple words can be processed as well. A digital voice recorder facilitates the addition of pronunciations. The system also supports a dual language mode for vocabulary training.

5.2. *The Dictionary*

As discussed in Section 2.1, words can be ordered by their probability of occurrence in the corpus. Let $|\mathbf{D}|$ be the total number of words in the training set. We want to partition the dictionary $\mathbf{D} = \{\mathbf{w}_1, ..., \mathbf{w}_{|\mathbf{D}|}\}$ into a set $\mathbf{M} = \{\mathbf{M}_1, ..., \mathbf{M}_{|\mathbf{M}|}\}$, $\mathbf{M}_m = \{\mathbf{w}_{m1}, ..., \mathbf{w}_{m|\mathbf{M}_m|}\}$ of modules, such that $\mathbf{M}_1 \cup \mathbf{M}_2 \cup ... \cup \mathbf{M}_{|\mathbf{M}|} = \mathbf{D}$ and the modules are disjoint. Also, the modules should be ordered by difficulty and probability with module $\mathbf{M}_1$ containing the most simple and most frequent words. For this purpose, we assign a difficulty $L_D(\mathbf{w}_j)$ to each word $\mathbf{w}_j$ and define a first ordering criterion for each word $\mathbf{w}_{mi}$ and $\mathbf{w}_{lj}$ in modules $\mathbf{M}_m$ and $\mathbf{M}_l$

$$m \geq l \Leftrightarrow L_{\mathbf{D}}(\mathbf{w}_{mi}) \geq L_{\mathbf{D}}(\mathbf{w}_{lj}). \tag{27}$$

To guarantee optimal progress during training the second ordering criterion follows probability. Let $P(\mathbf{w}_j)$ denote the *probability of occurrence of word* $\mathbf{w}_j$ (see Table 1). We sort by

$$m \geq l \Leftrightarrow P(\mathbf{w}_{mi}) \leq P(\mathbf{w}_{lj}). \tag{28}$$

Our analysis also found that these two objectives are correlated, because on average, more frequently used words are less difficult. We tried different com-

binations of the two objectives for the final sorting criterion $\tilde{L}_{\mathbf{D}}(\mathbf{w}_j)$ of our dictionaries and settled on

$$\tilde{L}_{\mathbf{D}}(\mathbf{w}_j) = \frac{L_{\mathbf{D}}(\mathbf{w}_j)^2}{P(\mathbf{w}_j)}. \tag{29}$$

The sizes of our modules range from 100 words for the simple ones and 250 words for the more advanced modules.

5.3. *Word Difficulty*

Our measure of the word difficulty $L_{\mathbf{D}}(\mathbf{w}_j)$ is based on the following considerations: First, we know from previous sections that, if $|\mathbf{w}_j|$ is the length of the word measured in letters, the average minimum number of bits needed to code this word is $H(X)|\mathbf{w}_j|$. Since the symbol entropy $H(X)$ is constant, it is safe to assume that the word difficulty is proportional to its length, that is

$$L_{\mathbf{D}}(\mathbf{w}_j) \propto |\mathbf{w}_j|. \tag{30}$$

Second and similar to color code optimization, there are dyslexic pairs which make certain words more difficult to learn for people with dyslexia. We call the letters belonging to a dyslexic pair a *dyslexic pitfall*. We define a function $\mathrm{pit}(x_i)$ for the letters x_i as

$$\mathrm{pit}(x_i) = \begin{cases} 1, \text{ if } x_i \text{ is dyslexic pitfall} \\ 0, \text{ else} \end{cases}, \tag{31}$$

$x_i \in \mathbf{A}$. Finally, "silent letters", which are written but not pronounced, pose an additional difficulty. They appear very frequently and in different combinations in English and French (see [42]), but can mostly be represented well by digrams. The function $\mathrm{sil}(x_k, x_l)$ defines a weight for these pairs

$$\mathrm{sil}(x_k, x_l) = \begin{cases} 1, \text{ if } (x_k, x_l) \text{ is silent letter pair} \\ 0, \text{ else} \end{cases}, \tag{32}$$

$\forall (x_k, x_l) \in \mathbf{A}^2$. We calculate the difficulty $L_D(\mathbf{w}_j)$ as

$$L_{\mathbf{D}}(\mathbf{w}_j) = |\mathbf{w}_j| + \sum_{i=1}^{|\mathbf{w}_j|} pit(x_{i,j}) + \sum_{i=1}^{|\mathbf{w}_j|-1} \mathrm{sil}(x_{i,j}, x_{i+1,j}). \tag{33}$$

Thus, the occurrence of a silent pair or dyslexic pitfall letter simply increases the perceived length of a word and eventually the number of bits to encode it, because a larger effort is needed to learn and remember the word.

5.4. *Word Selection and Error Entropy*

The purpose of the word selection controller is to select a word $\mathbf{w}_j$ from a module $\mathbf{M}_m$ in such a way that the user makes most progress. Progress, in turn, means to reduce the uncertainty of the knowledge about the words in $\mathbf{M}_m$. To this end, we define and utilize error entropy as a measure for the user's uncertainty and progress. An error entropy of 0 thus means that the user has learned the entire module. The objective of the controller is hence to *minimize error entropy*. To this end, we distinguish between symbol error entropy and word error entropy.

Symbol error entropy: We define a symbol error matrix or symbol confusion matrix P_C to monitor the symbol error probability. P_C is a $N \times N$-matrix, where $N = |\mathbf{A}|$. $P(x_k|x_l)$, $x_k, x_l \in \mathbf{A}$, is the *conditional probability* that a user enters erroneously x_k instead of x_l.

$$P_C = \begin{bmatrix} ... & ... & ... \\ ... & P(x_k|x_l) & ... \\ ... & ... & ... \end{bmatrix}, \quad x_k, x_l \in \mathbf{A}. \tag{34}$$

The columns of P_C partition unity. When the user decreases the number of errors over time through proper learning, P_C becomes the identity matrix. We initialize it with random numbers and a bias towards $P(x_l|x_l) = bias \leq 1$.

Let E be a binary random variable with $E = e_1$ indicating an error and $E = e_0$ being the correct answer. We define the error probability $P_E(x_l)$ for x_l as the probability that a user does not enter the correct letter x_l

$$\begin{aligned} P_E(x_l) = P(e_1|x_l) &= \sum_{k=1, k\neq l}^{N} P(x_k|x_l) \\ &= 1 - P(x_l|x_l). \end{aligned} \tag{35}$$

Now, the *global symbol error probability* $P_E(X)$ can be calculated as a weighted sum of the conditional errors.

$$\begin{aligned} P_E(X) = P(E = 1|X) &= \sum_{l=1}^{N} P(x_l) P_E(x_l) \\ &= 1 - \sum_{l=1}^{N} P(x_l) P(x_l|x_l). \end{aligned} \tag{36}$$

We are now in place to define a *conditional symbol error entropy* $H(E|X)$. It measures the residual

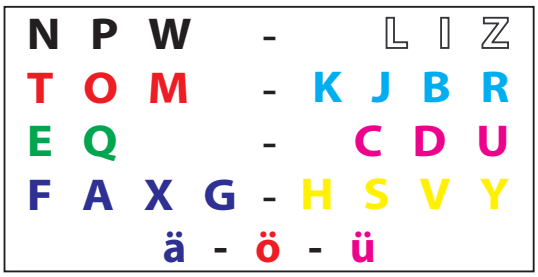


Fig. 14. Optimized color code for German.

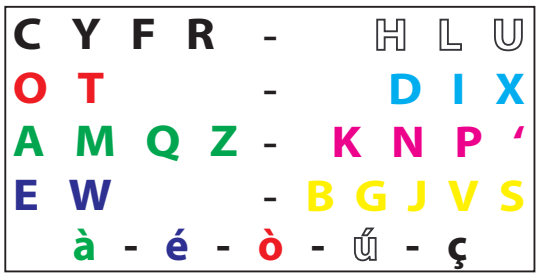


Fig. 15. Optimized color code for French.

uncertainty of any symbol of **A**. $H(E|X)$ can be expressed as the weighted sum of conditional entropies, where the conditional entropy is the error entropy of an individual symbol:

$$H(E|X) = -\sum_{l=1}^{N} P(x_l) \sum_{k=1}^{N} P(x_k|x_l) \log(P(x_k|x_l)). \quad (37)$$

The maximum $H(E|X) = \log(N)$ is reached when each column of P_C is uniformly distributed. The minimum $H(E|X) = 0$ is obtained if for each l there is a k such that $P(x_k|x_l) = 1$. Theoretically, this state can be achieved if a user constantly confuses a letter with another one. In this case, we know the confusion and no uncertainty is left. In practice, however, the minimum is reached for $P(x_l|x_l) = 1, \forall l$. *The fastest progress in learning is thus achieved by efficiently minimizing* $H(E|X)$.

Word error entropy: A second important aspect of error is related to words. For instance, the anagrams *"heart"* and *"earth"* will have equal influence on $H(E|X)$ while they pose different difficulties at the word level. There is extensive literature about word error probabilities from computational linguistics, speech recognition, automatic spell checking, etc (e.g. [43]). Most of them employ some sort of letter confusion, similar to $H(E|X)$. Therefore, we define word error entropy by the following variables.

Let D be the random variable accounting for events (words) from **D** and let the word error be a binary random variable E, as before. $P_E(D) = P(E|D)$ is the probability of a spelling error in a word of **D**. Thus $\forall \mathbf{w}_j \in \mathbf{D}$

$$P(E = e_1|D = \mathbf{w}_j) = P(e_1|\mathbf{w}_j) = 1 - P(e_0|\mathbf{w}_j). \quad (38)$$

We initialize $P(e_1|\mathbf{w}_j)$ for every word $\mathbf{w}_j = \{x_{1j}, ..., x_{|\mathbf{w}_j|j}\}$. When a user enters an answer $\tilde{\mathbf{w}}_j = \{\tilde{x}_{1j}, ..., \tilde{x}_{|\mathbf{w}_j|j}\}$ to the prompted $\mathbf{w}_j$, we compare $\tilde{\mathbf{w}}_j$ to $\mathbf{w}_j$ to obtain the number $N_E(\mathbf{w}_j, \tilde{\mathbf{w}}_j)$ of misspellings

$$N_E(\mathbf{w}_j, \tilde{\mathbf{w}}_j) = \sum_{i=1}^{|\mathbf{w}_j|} 1_{\{x_{ij} \neq \tilde{x}_{ij}\}}. \quad (39)$$

We essentially count the misspellings when the user types the word. $P(e_1|\mathbf{w}_j)$ is approximated by $N_E(\mathbf{w}_j, \tilde{\mathbf{w}}_j)$

$$P(e_1|\mathbf{w}_j) \approx \frac{N_E(\mathbf{w}_j, \tilde{\mathbf{w}}_j)}{|\mathbf{w}_j|}. \quad (40)$$

We finally define a *conditional word error entropy* $H(E|D)$ which measures the uncertainty of a word error over the dictionary **D** by

$$H(E|D) = -\sum_{j=1}^{|\mathbf{D}|} P(\mathbf{w}_j) \sum_{i=0}^{1} P(e_i|\mathbf{w}_j) \log(P(e_i|\mathbf{w}_j)). \quad (41)$$

The maximum entropy $H(E|D) = 1$ is reached when $P(e_0|\mathbf{w}_j) = P(e_1|\mathbf{w}_j) = 0.5, \forall j$. It is minimal, i.e. $H(E|D) = 0$, if either all letters are wrong or all are correct. In practice, of course, the former does not occur. *As before, efficient minimization of* $H(E|D)$ *guarantees fastest progress in learning.*

Cost function: The considerations from above suggest a cost function containing some linear combination of $H(E|X)$ and $H(E|D)$. To this end, we define

$$f_H = H(E|X) + \lambda_h H(E|D). \quad (42)$$

While conceptually more elegant, it has turned out in practice that a controller minimizing error entropy can occasionally delay the prompting of poorly learnt words. To this end we recommend to minimize a linear combination of $P(E|X)$ and $P(E|D)$ instead. The goal of the controller is to select the word that leads to a maximum decrease in the cost function - measured in the number of words and symbols to be prompted. Our implementation utilizes a greedy-style approach and defines a weight $\alpha(\mathbf{w}_j)$ for each word accounting for the symbol and word level errors:

$$\alpha(\mathbf{w}_j) = \lambda_j P(e_1|\mathbf{w}_j) + \frac{1}{|\mathbf{w}_j|}\sum_{i=1}^{|\mathbf{w}_j|}(1 - P(x_{ij}|x_{ij})) \quad (43)$$

$$\lambda_j = |\mathbf{D}| \cdot P(\mathbf{w}_j). \quad (44)$$

Using $P(\mathbf{w}_j)$ in λ_j enforces more frequent words to be prompted earlier, while employing $|\mathbf{D}|$ normalizes the two terms adequately. Again, in practice we minimize word error within the module $\mathbf{M}_m$ while the symbol error is minimized across all modules. The selection criterion for the next word $\mathbf{w}_i$ thus yields to

$$\alpha(\mathbf{w}_i) = \max\{\alpha(\mathbf{w}_j)|\mathbf{w}_j \in \mathbf{M}_m = \text{current module}\}. \quad (45)$$

The modules are switched when $P(E|D)$ falls below a threshold.

6. Results

This section presents a summary of the most important results achieved with our method and training system. We will give more examples for color codes we computed for German and French and summarize the experimental evaluation we have carried out to validate the effectiveness of the training. A detailed presentation of the results as well as a psychological interpretation is given in [44].

6.1. *Computation of Color Codes*

Fig. 14 and Fig. 15 present the results of our color code optimizations for German and French. We observe that the mappings are quite different for both languages. To validate the quality of the results we optimized codes for each of the four objectives of E separately and compared the residual energies with the results of the Pareto optimization. For instance the uniform color cost E_C is optimized up to about 2.98 bits entropy for each language compared to the maximum of 3 bits for a fully uniform color distribution. The frequent letter pairs cost E_{M_1} is such that in English, for instance, the 35 most frequent letter pairs are mapped to different colors and that for all three languages, less than 4.7% of all occurring pairs have the same color. The dyslexic pair cost E_D of the examples considers all dyslexic pairs in all three languages and their color distances are considerably larger than the average color distance of all symbols. For English we achieved 207.9 compared to 159.4 on average. Finally as for E_U, 896 out of 8000 words are mapped to the same code in English. On average, 2.25 words are assigned to the same code and no more than 6 words.

Fig. 16. A training session at ETH.

6.2. *Experimental Evaluation and Psychological Study*

In order to validate the presented recoding method, we carried out an extensive user study at ETH Zurich over a period of 6 months in collaboration with the Institute for Neuropsychology of the University of Zurich. The goal was to prove the effectiveness of our method. The user groups involved 43 German speaking children aged 9 to 11 with developmental dyslexia and 37 matched children with normal reading and writing skills.

We divided them into 4 different groups. A group of children with developmental dyslexia (DW) and a control group (CW) both practiced with the training software during a first period of 3 months and for 15-20 minutes four times a week amounting to a total of about 800 minutes of interactive training. The second dyslexic group (DO) and control group (CO) received no training. During a second, crossover period the conditions were swapped and the groups DW and CW had to suspend training.

We selected a subset of 1500 words from our dictionary $\mathbf{D}$ based on our definition of importance. The set was additionally tuned by elementary school teachers and psychologists. We measured the children's writing amelioration by a dictation containing 100 words. A random half of the words were used during the training session and the second half served for testing the children's ability to generalize to novel words. The two word sets were carefully matched according to frequency and difficulty, as determined by the ECI German 3 corpora. In addition, they were matched with respect to the number of syllables. Our analysis showed that the errors between the learned and the new words correlated

Table 6
Relative improvements in dictation error. (p = paired T-Test for error-sum from spring to winter.)

	spring to summer			spring to winter			
	all words	learned words	new words	all words	learned words	new words	p
CO	17			50	56	57	0.047
CW	27	27	26	34	31	34	0.000
DO	4			31	38	26	0.000
DW	27	32	23	25	23	24	0.000

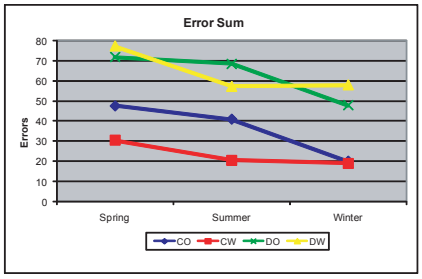


Fig. 17. Improvements in dictation error for all participating groups.

with r=0.93. All children had to pass our writing test before training, after three months and at the end of the study.

The system was utilized exactly as described in the paper, but we added a simple aging term to the controller to guarantee long term repetition of learnt words. In the used implementation, the words were put into a repetition module $\mathbf{M}_R$ and prompted again after a certain period. In addition, the color game had to be played during the first 5 minutes of training for the first two weeks. We also extended the software to launch with memory stick only and to write each interaction to the stick along with a time stamp.

At the beginning of the study, each child underwent a series of standard psychological tests. These included classical German writing (Salzburger-Lese und Rechtschreibtest SLRT, Diagnostischer Rechtschreibtest für fünfte Klassen DRT5 and a German reading test (Zürcher Lesetest ZLT) to quantify writing and reading errors, and a standard German intelligence test HAWIK III to exclude children with an IQ lower than 85. In addition, we carried out an attention test to exclude children suffering from attention-deficit-hyperactivity disorders (ADHD/ODD-Elternfragebogen), a categorization test to measure possible planning problems (MWCST) and a handedness performance test to measure hand performance skill (Hand-Dominanz-

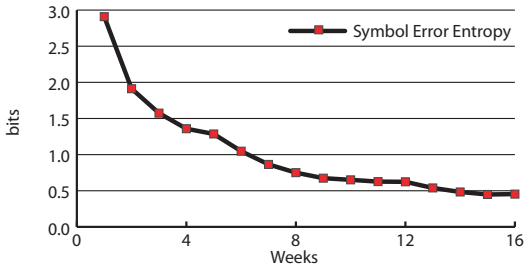


Fig. 18. Error entropy H(E,X) of one of the subjects of DW as a function of training time.

Test). The average IQ of all children was 107 and ranged from 87 to 132. The children used their home PCs for training, but convened weekly in a computer laboratory at ETH for supervision by student helpers, to download data, and to ask questions. Fig. 16 shows a snapshot of a training session at ETH.

The results of our study are summarized in Table 6 which presents the achieved error reduction of the groups between the dictations before and after training. We observed a significant improvement of 27% on average of the writing skills of the children with dyslexia **DW** after training as opposed to the 4% achieved by their counterparts **DO** without training. 1/3 of the **DO** group did not improve at all. This proves the effectiveness of our method.

As a further significant finding, the **DW** group improved by 32% on words from the learnt subset, but also about 23% on the generalization dataset. This result leads to the conclusion that the recoding can effectively generalize to new, unknown words, a highly desired property. Finally, compared to non-dyslexic children, the groups **CW** and **CO** improved by 27% and 17% respectively. Fig. 17 gives a graphical summary of the results.

Especially the similar results of **CW** and **DW** evidence that children with dyslexia can on average achieve similar learning performance than their counterparts, if the presented information is conveyed through undistorted perceptual pathways — which is the very design principle of our method. Finally, Fig. 18 exemplifies how the error entropy according (37) of one subject of the group **DW** declines as a function of training time.

7. Discussion

We presented a novel framework for the multimodal representation of words and demonstrated

its utility as an effective learning aid for people with dyslexia. The method recodes an input string into a set of spatial, color, shape, and auditory codes which altogether reroute information through multiple perceptual cues. The entire framework is based on statistical modeling of language and on the fundamental principles of information theory. The experimental validation of our method has clearly shown its high effectiveness and gives empirical evidence for the suitability of the chosen model.

An apparent limitation of the current concept is the limited emphasis of the phonological structure of a word and of explicit phoneme-grapheme mappings. In particular results from speech processing suggest, for instance, that such mappings are central to reliable speech recognition. In dyslexia research, however, the mental processes and mappings underlying language acquisition are still less well understood. While phoneme-grapheme mapping is generally considered a central problem in dyslexia, it is less clear to what extent explicit representations of phoneme-grapheme mappings will lead to superior results than our current syllable-based topological structures. In addition, the improvements achieved in our study give sufficient empirical evidence that the current scheme works very well.

We are currently performing a major data analysis of all data records of our experimental evaluation. Early results suggest a mildly better correlation between word difficulty and number of errors in a word when considering phoneme-grapheme separations explicitly. Whether this will eventually result in a refined learning scheme is currently not clear. Initial considerations, however, have turned out that phoneme-based word decomposition does not necessarily play well with the concept of topological coding and would require a major redesign of the method. We will definitely explore this issue further in future scientific work.

We also have to investigate synchronization of the musical and acoustic representations to better address potential timing deficits in dyslexics. Another potential limitation relates to the control condition of the experiment. Ideally, one would want to compare the achieved results to a control group with conventional additional training. However, since such conditions are virtually impossible to realize, comparison to untrained groups have become a widely accepted scientific standard in dyslexia research.

The data analysis of the user interactions will also provide more insight into the process of learning and into the rate-distortion behavior of our model. In addition, we plan to design Bayesian networks to infer different types of error, such as typo, color error, etc, and to automatically switch between individual games.

Acknowledgment

We thank Lutz Jäncke, Monika Kast and Martin Meyer for assisting and advising us on the user study; Basil Weber and Ghislain Fourny for their support in programming the Dybuster prototype; Doo-Young Kwon for designing a part of the user interface; all children and parents for participating in the user study; and the Swiss Dyslexia Association for helping us to recruit them.

References

[1] M. Snowling, Developmental dyslexia: A cognitive developmental perspective, in: P. G. Aaron, R. M. Joshi (Eds.), Reading and Writing Disorders in Different Orthographic Systems, Vol. 52 of NATO ASI series. Series D, Behavioural and Social Sciences, Kluwer Academic Publishers, 1989, pp. 1–23.

[2] P. Reitsma, Orthographic memory and learning to read, in: P. G. Aaron, R. M. Joshi (Eds.), Reading and Writing Disorders in Different Orthographic Systems, Vol. 52 of NATO ASI series. Series D, Behavioural and social sciences, Kluwer Academic Publishers, 1989, pp. 51–74.

[3] M. B. Denckla, A neurologist's overview of developmental dyslexia, in: P. Tallal, et al. (Eds.), Temporal Information Processing in the Nervous System with Special Reference to Dyslexia and Dysphasia, Vol. 682 of Annals New York Academy of Sciences, 1993, pp. 23–26.

[4] M. Coltheart, Lexical access in simple reading tasks, Strategies of Information Processing (1978) 151–216.

[5] F. Ramus, Developmental dyslexia: specific phonological deficit or general sensorimotor dysfunction?, Current Opinion in Neurobiology 2003 13 (2003) 212–218.

[6] P. H. Wolff, Impaired temporal resolution, in: P. Tallal, et al. (Eds.), Temporal Information Processing in the Nervous System with Special Reference to Dyslexia and Dysphasia, Vol. 682 of Annals New York Academy of Sciences, 1993, pp. 87–103.

[7] J. Everatt, Reading and Dyslexia: Visual and attentional processes, Routledge, 1999.

[8] K. Pammer, W. Lovegrove, The influence of color on transient system activity: Implications for dyslexia research, Perception & Psychophysics 63 (3) (2001) 490–500.

[9] P. Tallal, Improving language and literacy is a matter of time, Nature Reviews 5 (2004) 721–728.

[10] B. A. Wright, L. J. Lombardino, W. M. King, C. S. Puranik, C. M. Leonard, M. M. Merzenich, Deficits in auditory temporal and sepctral resolution in lanuage-impaired children, Nature 387 (1997) 176–178.

[11] M. Habib, The neurological basis of developmental dyslexia: An overview and working hypothesis, Brain 123 (2000) 2373–2399.
[12] J. Rüsseler, Neurobiologische Grundlagen der Lese-Rechtschreibschwäche: Implikationen für Diagnostik und Therapie, Zeitschrift für Neuropsychologie 17 (3) (2006) 101–111.
[13] M. Spitzer, Nervensachen: Geschichten vom Gehirn, Suhrkamp, 2005.
[14] P. Tallal, N. Gaab, Dynamic autiory processing, musical experience and language development, Trends in Neuroscience 29 (7) (2006) 382–390.
[15] K. Overy, R. I. Nicolson, A. J. Fawcett, E. F. Clarke, Dyslexia and music: Measuring musical timing skills, Dyslexia 9 (1) (2003) 18–36.
[16] T. Kujala, K. Karma, R. Ceponiene, S. Belitz, P. Turkkila, M. Tervaniemi, R. Näätänen, Plastic neural changes and reading improvement caused by audiovisual training in reading-impaired children, Procedings of the National Academy of Sciences of the United States of America 98 (2001) 10509–10514.
[17] J. Tijms, J. Hoeks, A computerized treatment of dyslexia: Benefits from treating lexico-phonological processing problems, Dyslexia 11 (2005) 22–40.
[18] Webpage of the international dyslexia association, Webpage.
URL `www.interdys.org`
[19] Webpage provided by iansyst ltd., a reseller for dyslexia products, Webpage.
URL `www.dyslexic.com`
[20] Various, different software programms, see for instance.
URL `www.legasthenie-software.de, www.dyslexic.com`
[21] J. Strydom, S. du Plessis, The Right to Read: Beating Dyslexia and other Learning Disabilities, Remedium Publisher, 2000.
[22] R. D. Davis, E. M.Braun, The Gift of Dyslexia: Why Some of the Smartest People Can't Read and How They Can Learn, 1st Edition, Perigee Books, 1997.
[23] D. Jurafsky, J. Martin, Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics, Prentice-Hall, 2000.
[24] T. Cover, J. Thomas, Elements of Information Theory, John Wiley & Sons, Inc., 1991.
[25] C. Ware, Information Visualization. Perception for Design., 2nd Edition, Morgan Kaufmann Publishers Inc,US, 2004.
[26] E. R. Tufte, The Visual Display of Quantitative Information, 2nd Edition, Graphics Press, 2001.
[27] M. Gross, Visual Computing, Springer, 1994.
[28] P. Baldi, L. Itti, Attention: Bits versus wows, in: M. Zhao, Z. Shi (Eds.), Proceedings of the International Conference on Neural Networks and Brain (ICNN&B05), Vol. 1, IEEE Press, Beijing, China, 2005.
[29] C. H. Lee, A. Varshney, D. Jacobs, Mesh saliency, ACM Transactions on Graphics (Proceedings of SIGGRAPH 2005) 24 (3) (2005) 659–666.
[30] British National Corpus, Website (2004).
URL `www.natcorp.ox.ac.uk`
[31] European Corpus Initiative Multilingual Corpus I (ECI/MCI), Website.
URL `www.elsnet.org/resources/eciCorpus.html`
[32] G. K. Zipf, Human Behaviour and the Principle of Least Effort: An Introduction to Human Ecology, Addison-Wesley, 1949.
[33] G. Altmann, G. Wimmer, The theory of word length, in: P. Schmidt (Ed.), Issues in general linguistic theory and the theory of word length, Vol. 15 of Glottometrika, 1996, pp. 112–133.
[34] D. E. Knuth, The TeXbook, Addison Wesley Publishing Company, 1986, in Chapter H.
[35] T. Bell, J. Cleary, I. Witten, Text Compression, Prentice Hall, 1990.
[36] C. E. Shannon, The Mathematical Theory of Information, University of Illinois Press, 1949.
[37] D. Benoit, E. D. Demaine, J. I. Munro, V. Raman, Representing trees of higher degree, in: Proceedings of the 6th International Workshop on Algorithms and Data Structures (WADS'99), LNCS, Vol. 1663, 1999, pp. 169–180.
[38] G. Wyszecki, W. S. Stiles, Color Science: Concepts and Methods, Quantitative Data and Formula, 2nd Edition, John Wiley & Sons, 1982.
[39] C. Glasbey, G. van der Heijden, V. Toh, A. Gray, Colour displays for categorical images, Color Research & Application 32 (4) (2007) 304–309.
[40] M. Ehrgott, Multicriteria Optimization, Springer, 2000.
[41] Website of the eth spin-off dybuster, Website.
URL `www.dybuster.com`
[42] M. Schneider, Laut-Buchstaben-Beziehungen im Deutschen.
URL `URL: www.schneid9.de/pdf/lautbuchstaben.pdf`
[43] E. Brill, R. C. Moore, An improved error model for noisy channel spelling correction, in: Proceedings, 38th Annual Meeting of the Association for Computational Linguistics, 2000, pp. 286–293.
[44] M. Kast, M. Meyer, C. Voegeli, M. Gross, L. Jaenke, Computer-based multisensory learning in children with developmental dyslexia, Restorative Neurology and Neurosciences, 2007.