Quantifying Visual Feature Detection in Word Identification Across Vocabulary Sizes

Carolyn Yao
Stuyvesant High School
345 Chambers Street
New York, NY 10282

Mentor: Professor Denis Pelli
Psychology and Neural Science
6 Washington Place, Room 959
New York University
New York, NY 10003

Research Advisor: Jonathan Gastel
Stuyvesant High School
345 Chambers Street
New York, NY 10282

Abstract

When we look at an image, our visual system breaks down the image into features. Each feature is an independently detected, discrete component of the image (Robson and Graham, 1981; Pelli, Farell, and Moore, 2003). Vision combines the features to identify the object. Instead of looking at simple objects, like gratings and plaids that were studied in the past, we explore the role of features in word identification. Words are richer stimuli that allow more profound experiments. In particular we study the effect of the number of possible words on the observer's identification of one. Such context effects are very important in everyday vision. We extended the well-known standard "probability summation" model for object detection to identification. We assume that the observer correctly identifies an object when she detects a number *k* of its features or can guess correctly when fewer than *k* features are detected. We use estimate the observer's *k* from measurements of proportion correct as a function of duration of presentation of the word. A random four-letter-word from a vocabulary set is flashed for the observer in various short durations using our own MATLAB program. This is done separately with three different word sets, containing 10, 26, or 1708 words. From the measured human performance, *k* was found to grow logarithmically with the number of words in the set. Identifying one of *n* words requires $\log_2 n$ bits of information. Our results show that each feature provides 1.7 bits of information about which word is present. 1.7 bits corresponds to distinguishing 3 values, as opposed to past research which was unable to prove that a feature could contain more than 1 bit, corresponding to 2 values: present or absent. These results help us better understand how we recognize words and how the ability to identify objects varies with the number of possible alternatives. Our findings apply to reading, to understand the limits to reading speed and comprehension, and also apply to possibly optimizing text design to facilitate visual processing.

The number of features used to identify a word

Introduction

What does it mean to perceive something? How do we piece together visual parts to obtain information about our world and identify what we see? We aim to examine the effects of familiarity and number of alternatives on identification of objects, in this case English words.

Much like cells are the building blocks of life, features are the most basic components in seeing an image. Features are detected independently of each other (Robson and Graham, 1981; Pelli, Farell, and Moore, 2003). The first stage of vision in the brain is feature detectors (Hubel and Wiesel, 1962), but the next stages that combine those features are less clear. Objects vary in the number of features they contain, but observers usually don’t need all the features to identify. The number of features required to identify an object depends on the task. In this paper we look at features in the context of identifying words. It is important to distinguish psychological features in any image from typographic characteristics of letters, such as font, color, size, or orientation. Research on vision has not yet produced a catalog of all the features used in human vision, but it is well-established that a gabor is one of them. A gabor is a striped disk with soft edges (Fig.1). It can take on any position and orientation (tilt). Any image may be composed of any number of gabors with different

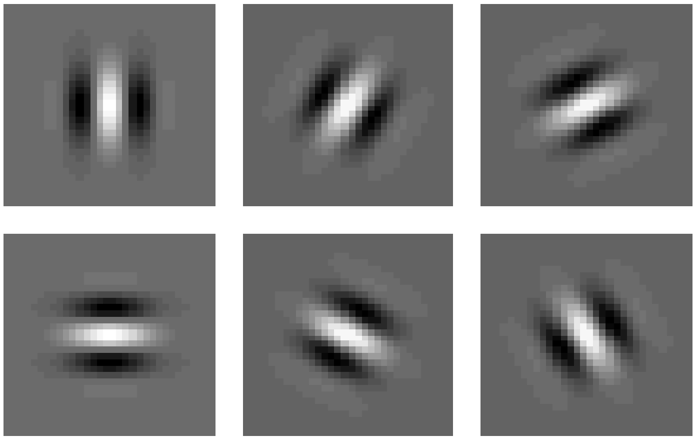


Figure 1: the six even-symmetric gabor filters (Kumar, 2012) in their respective boxes have different orientations, and are essentially six different features.

orientations, and those gabors can be detected independently as the smallest discrete components. The identifiability of any image can be degraded by presenting it very briefly (Massaro & Hary1986). It is thought that word recognition is mediated by letter identification, and that letter identification is mediated by feature detection (Gough, 1984; Massaro, 1984; Paap, Newsome, & Noel, 1984; Pelli, Farell, & Moore, 2006). To model word identification, we first look at the detection of its features.

We start with the probability summation model for visual detection. Suppose the word has *n* features. Extending detection to identification, we assume that an observer will identify an image whenever at least a certain number *k* of *n* features is detected or by chance, the observer guesses correctly with fewer than *k* features. To simplify the modeling, we suppose that all features are detected with equal probability. Here is a complete derivation of the identification model starting from detection, in four equations, with thanks to Suchow and Pelli. Feature detection is a Poisson process. Suppose that in one glimpse the observer has probability of 1-1/*e* of detecting a given feature. If the time for one glimpse is *τ* (tau) and *T* is total duration, then in the whole presentation, the observer will have time for *T*/*τ* glimpses. Given that the glimpses are independent, the probability of detecting at least once in the interval is

$$p = 1 - e^{-T/t} \tag{1}$$

This is the probability of one specific feature being detected. Words have many features, so now we consider the probability of several features being detected.

Each feature is either detected or not. Thus, we can make the analogy of features to weighted coins, and the chance of detection to the chance of flipping a head. The probability of flipping a certain number of heads was worked out by the Swiss mathematician

Jacob Bernoulli (1654 - 1705). A Bernoulli Process is a specific case of the Poisson process. For each feature, the probability of detection *p* corresponds to flipping a weighted coin that lands heads up. The binomial probability p_i of exactly *i* heads among *n* coins is:

$$p_i = \binom{n}{i} p^i (1 - p)^{n-i} \tag{2}$$

In our application, this is the probability of detecting exactly *i* features out of the total number of features, *n*. The probability of identification P_i, big P, is the probability of detecting at least *k* features, p_i plus the probability *g* of guessing blindly when not enough features are detected. We imagine the graph of p_i, deeming the area of the graph between *k* and *n* values as identifying the object, and anything in the interval 0…*k*-1 values as failing to identify an object.

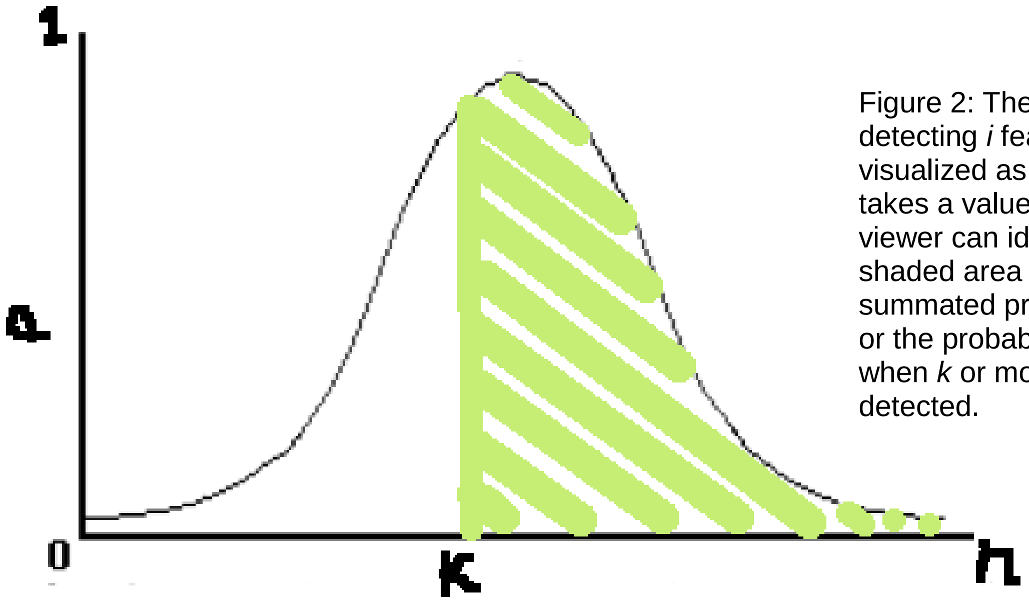


Figure 2: The probability of detecting *i* features out of *n* is visualized as a bell curve. When *i* takes a value of *k* or larger, the viewer can identify the object. The shaded area represents the summated probability of *p(k)* to *p(n)*, or the probability of identification when *k* or more features are detected.

So we get find Eq. 3 below, where *P*, the identification probability, is the probability of detecting enough features (i.e., not detecting *k*–1 or fewer) and the probability of guessing the object correctly when not enough features are detected.

$$\begin{aligned} P &= 1 - \sum_{i=0}^{k-1} p_i + g\sum_{i=0}^{k-1} p_i \\ &= 1 - (1-g)\sum_{i=0}^{k-1} p_i \end{aligned} \tag{3}$$

When *i*=1, we can rewrite *k*-1 as 0, and imagine the probability of detecting 0 features as failure to detect repeated *n* times. We substitute 1-*p* with the event rate from Eq. 1 and end up with *T*, *n*, and *τ* in our final equation, merging the Poisson and Bernoulli processes:

$$\begin{aligned} P_{k=1} &= 1 - (1-g)p_0 \\ &= 1 - (1-g)(1-p)^n \\ &= 1 - (1-g)e^{-nT/\tau} \end{aligned} \tag{4}$$

Eq. 4 shows that when *i*=1 the performance depends on the number *n* of features and the duration –T/*τ* solely through their product, the event rate. Eq. 4 does not apply when *i* takes the general *k*, because we do count multiple detections across features but don’t count extra detection over time of the same feature. Performance still depends approximately on just the detection rate *τ*, the guessing rate *g*, and the required number *k* of features.

Although we have these equations, the number of features we use to identify a complex visual object remains mostly unknown. But now that we’ve worked out a theory, which assumes that identification requires the detection of a certain number of features, and that features are detected independently over time, we know the probability of identification will grow as a binomial function at a rate determined solely by the number of features required, *k*. Thus, measuring the proportion of correct identifications as a function of duration should reveal the number of features used by the observer. The accuracy of this method was confirmed in the past using specific cases where the number of features used was already known.

Probability of identification increases with log duration, and the steepness of the curve largely depends only on *k*, the number of features required to identify the image. In the Admoni and Pelli 2004 paper on counting features, observers were asked to identify “Indy letters” that consisted of several gabor patches with 1, 2, or 4 gabors. Each gabor has two useful values, either horizontal or vertical (0 or 1), so IndyOne had two “letters”, IndyTwo had four, and IndyFour had sixteen. The proportion of correct identification rises more steeply for patches with more “letters”. The model fits the human performance well for both observers with the number *n* of features being equal to the number of gabor patches.

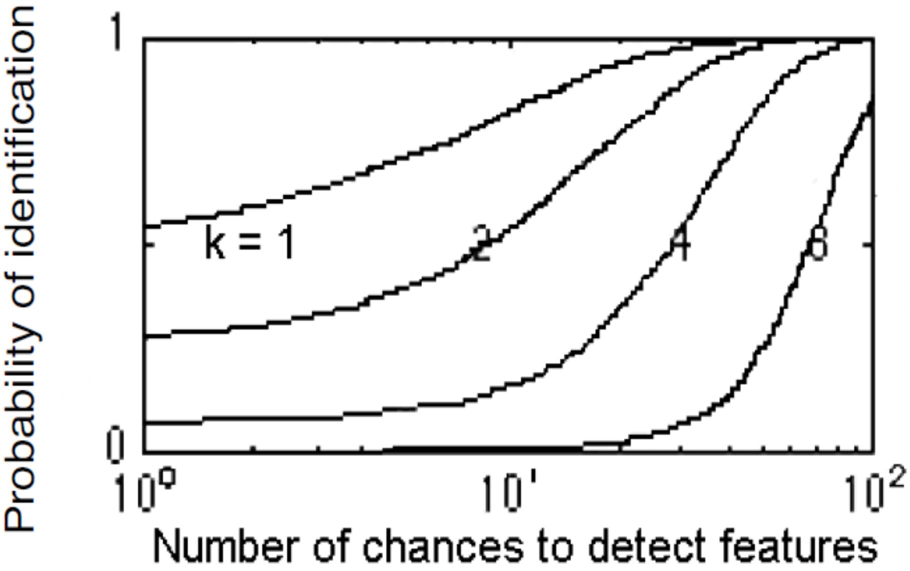


Figure 3 shows P_i as a model function of the number of "glimpses" afforded to the observer (T/τ) vs. the theoretical probability of identification. The steeper the slope, the more features *k* are needed to identify.

By just measuring proportion of correctly identified objects as a function of time or the T/τ chances to detect features (Admoni & Pelli 2004), we find the corresponding slope, and ultimately obtain the value of *k* of letter identification.

The story for complex objects is that the number *n* of features cannot be counted directly. Objects that are foreign to us especially elude this model, but would also present opportunity for research on identifying pure images rather than words, which are no longer considered "images" but text—pictures that have become a medium of mass communication. However, this model function makes it possible for us to compare word identification against identification of simple objects that have fit the function. We explored letter and word identification by driving up the number of identification alternatives for identifying, introducing a variable to the process in order to find out how features function when reading text.

Methods and Procedure

Observers:

The 15 participants in the experiment were between the ages 13 and 25, had normal-to-corrected acuity, normal contrast sensitivity, were English-proficient, and were naïve to the purpose of the experiment.

Recruitment:

Subjects were recruited if they met the above qualifications. Most were high school students. The subjects were asked to spend up to twenty minutes on a visual experiment and then asked if they had normal vision.

Stimuli:

The experiment was conducted on a MacBook, with brightness luminance controlled at 50 cd/m^2, which is the middle of the laptop's brightness range (Pelli, Burns, Farell, Page, 2006). The images were produced on the screen using the programming language MATLAB (MathWorks), and the external Psychophysics Toolbox (Brainard, 1997; Pelli 1997), which includes a variety of functions that cater to vision research. The observer's viewing distance was 50cm. The contrast was originally set at 0.1, but, after a few trials runs was reduced to 0.03 to bring performance down below the 100% ceiling.

The observer first saw a gray screen. Then, in a centered light gray box, several words flashed by, one word at a time, very quickly. After the subject clicked to signify readiness, the run began. The observer fixated on the center of the display with the guide of four orthogonal lines forming a crosshair. Black text as well as the program's speech offered instructions. Each run consisted of 60 trials, where a random 4-letter word from a bank of 10, 26, or 1708 words flashed. The 1708-word bank consisted of most four letter words in the English language. The 10 and 26 word contained random 4-letter words pulled from the 1708 word bank. For all sets, the x-height of the text was 0.5 degree of visual angle. The words flashed for one of several durations. Each of the durations was set as a condition and was defined in terms of the number of frames per second. We used 1, 2, 4, 8, 16, 32 frames per second, all powers of 2. They translated into 17ms, 33ms, 66ms, 132ms, 265ms, and 530 ms, respectively. The durations in this range successfully produced chance performance on the shortest durations, and close to perfect performance on the longest durations.

Task

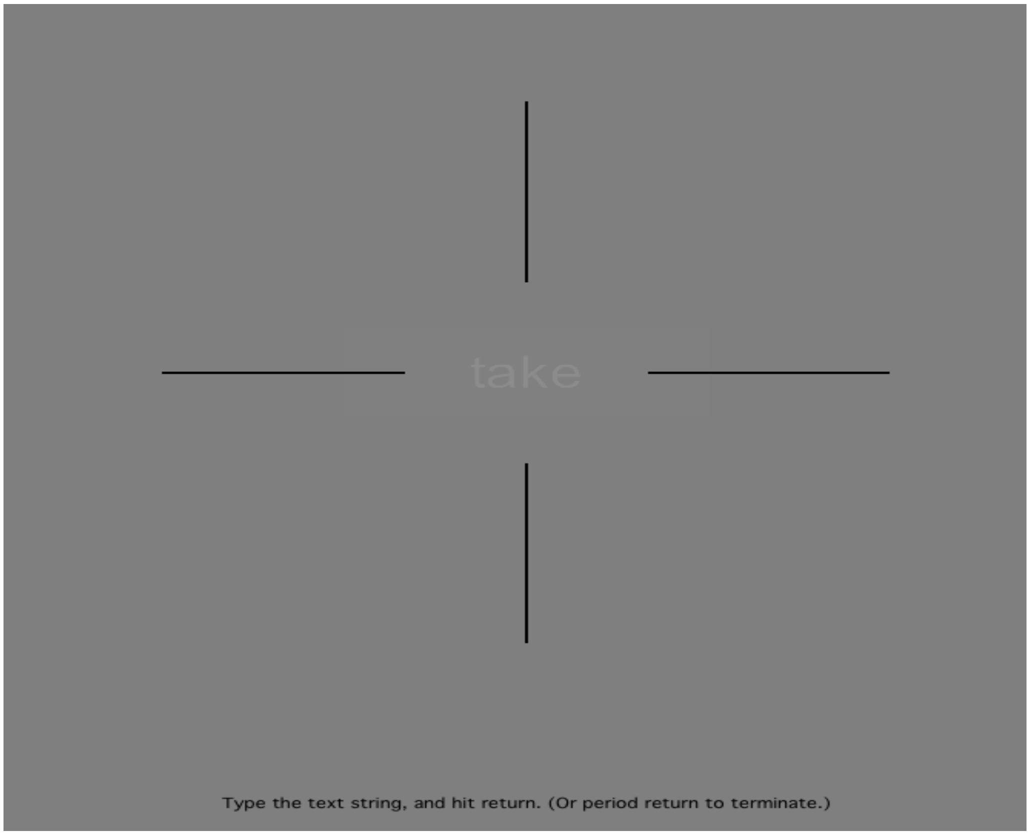
Before the 10-word and the 26-word sets, the observers gained familiarity of the words or alternatives they had for identifying in the upcoming run. For example, right before the 10-word run they would be given:

aunt
took
anne
fear
grin
much
huts
tell
camp
else

as possibilities to guess if they cannot make out the word completely, especially during the shortest durations.

For the 1708-word set, which contained most, if not all 4-letter words, observers did not need to familiarize within a list and went straight into the task at hand.

The observer, after each word, used the keyboard to enter the word they saw. If the input was part of the bank, the program registered it as either correct or incorrect, emitting a low-pitched hum if the answer was incorrect or rewarding a high-pitched beep for being correct. If the input was not part of the particular word bank (not a legal response), the program offered an alternative on the list closest to the input before scoring the initial guess as correct or incorrect.



Type the text string, and hit return. (Or period return to terminate.)

Figure 4 (right): the stimulus drawn with light gray against a fixed contrast gray screen using a function called DrawText in the font Arial. In this case, the word flashing by is “take” and is fixated in the center with the aid of the orthogonal lines. Directions for the observer are displayed on the bottom.

Learning curve:

A learning curve is an observer’s efficiency for identifying objects or characters from a new alphabet. After a fair amount of exposure, the learning curve slows immensely. In other studies, this aspect is accounted for with a practice run. However, in this experiment, the alphabet was not new—all stimuli come from the English alphabet. The subjects have already reached the plateau height of their learning curve.

Mechanisms of the program used to run the experiment:

Within the program, we created many variables to represent input, output, and controlled settings such as contrast, letter size, etc. The code was written so that the durations would occur in a random order and that the longest duration would guarantee a proportion correct of at least 0.99. In outputting the data based on the input of the observer, the program was able to calculate proportion correct easily; it did this by counting the number of correct responses and then dividing to find performance across the various durations.

Analysis

Once we had the empirical probabilities across durations, we used another MATLAB program, countFeatures, to draw curves of different steepness and to calculate the values of *k* across the runs. The program takes the data points and finds the value of *k* in Eq. 3 that fits the data the best (i.e., it minimizes the mean squared error between the model's predictions and the data).

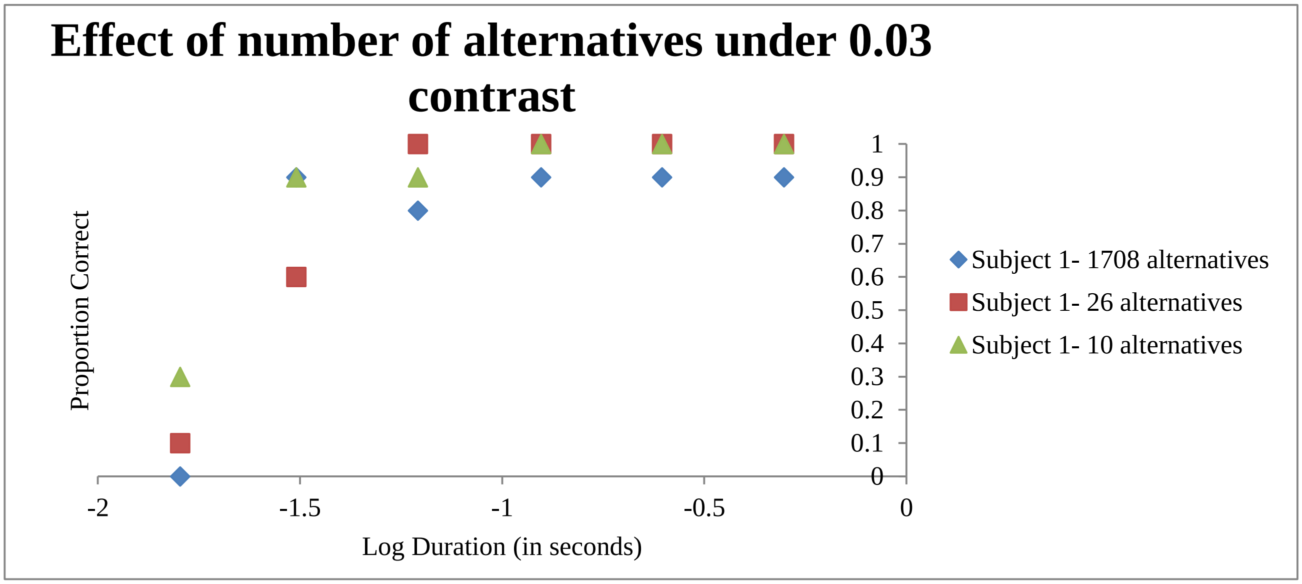


Figure 5 displays the raw data before the parameters of Eq. 3 are applied. These are data points for one individual across the three sets of alternatives. As it is, this graph gives us only an idea of how quickly the performances in each set of alternatives grow from shorter durations to longer ones.

Taking the data points into account, countFeatures graphs a curve that most closely follows the data and gives us the *k*, *τ*, and *n* values. The image below depicts the program's output: three curves graphing Eq. 3 that fits our human data.

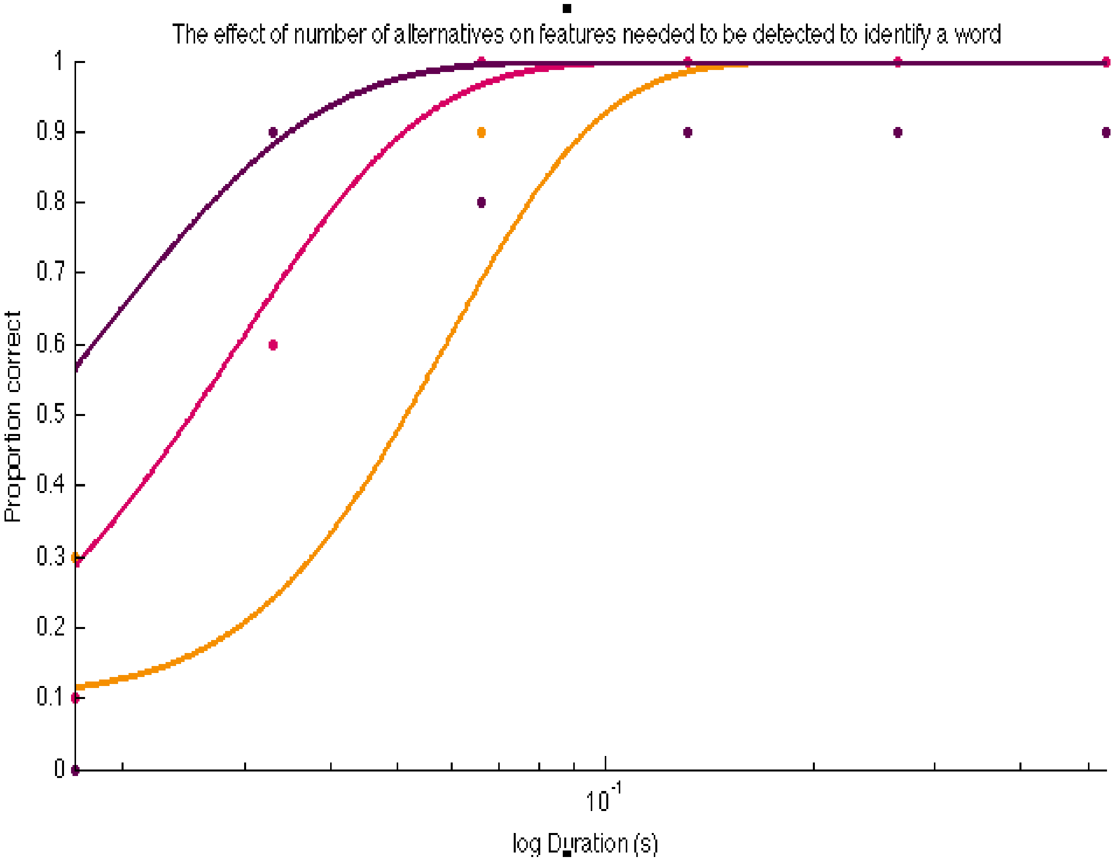


Figure 6: This graph includes the same set of data from Fig. 5 but account for the parameters in Eq. 3 and gives us curves that become steeper as they shift to the right, along with the *n, k,* and *τ* values that produce the curves.

Results

At first under our original 0.1 contrast, our duration vs. performance graphs could not produce slopes, as proportion correct hovered around 0.9 for all the durations, across all the word sets.

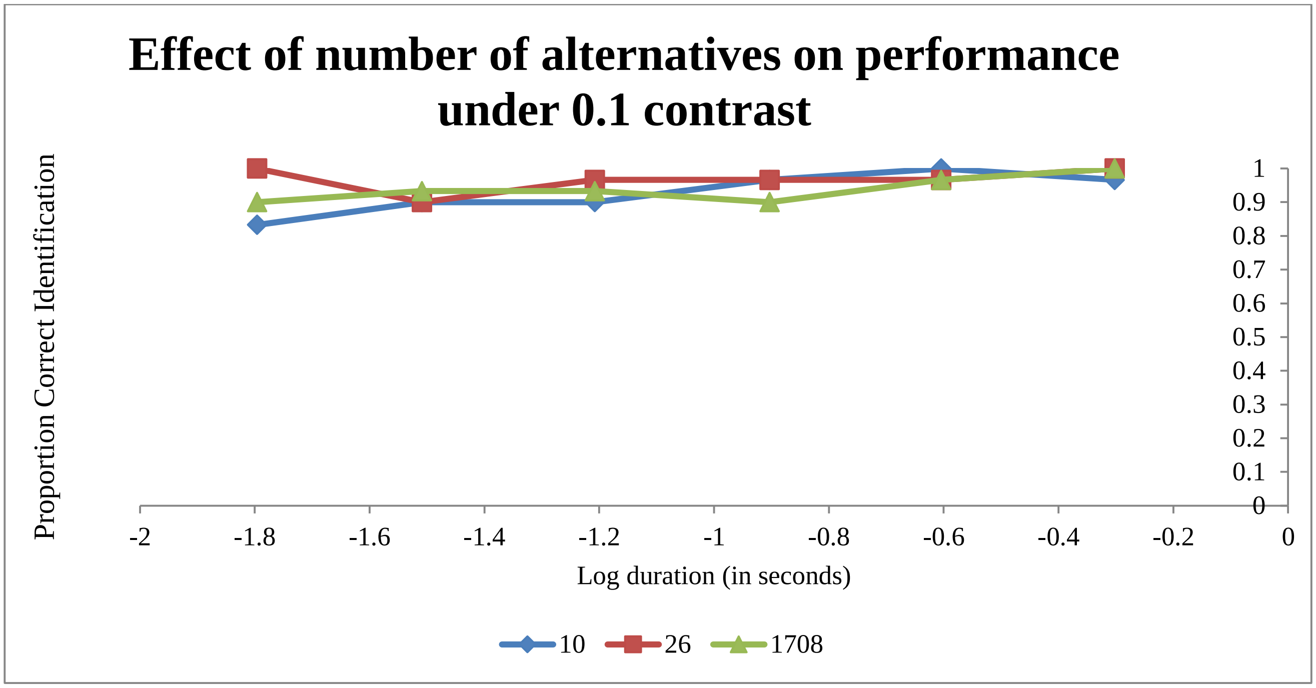


Figure 7: This displays the performances as a function of log duration across the three runs. Performance is at ceiling, near 100%, so there is hardly any difference in performance between the runs.

We lowered the fixed contrast to create more disparity in performance between the durations to end up with better curves. Longer durations would simply give more time to see in low contrast. It makes sense—if we degrade letter stimuli with low contrast, then it functions more as an image and less as a text. We separately tested one observer, using 10, 26, and 1708 words.

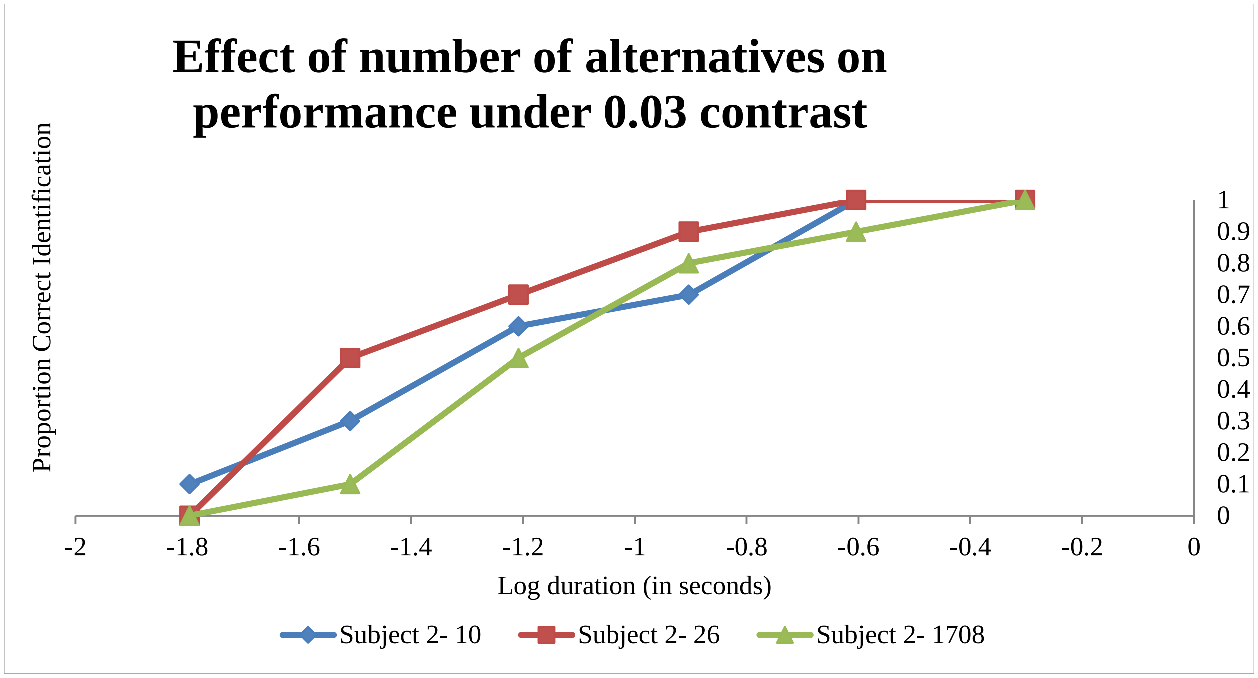


Figure 8 is a graphical presentation of what occurred when we reduced the contrast to 0.03, and increased in the number of alternatives. The shorter durations had performance levels near 0, whereas the longer durations gave ceiling performance, as desired.

The curves are much more defined and the slopes between the three conditions are much more discernable. And so we went onto using the method for many observers.

countFeatures gave us the n, τ, and k values for each observer. The n and τ values differed across the subjects because n notes the number of total features "available" to the specific observer, based on factors such as sensitivity to contrast, brightness level, size of critical detail, and durations afforded to them (Cobb and Moss, 1928). τ is subject to those factors as well.

Parameter	Mean across observers
n	7.31 ± 0.13
τ	.080 ± 0.01

Looking at all the subjects, we arrive at the crux of our results by averaging the k values in each set of alternatives across the subjects to find the mean *k* in each condition.

Number of alternatives (N)	Bits ($\log_2 N$)	k	Bits per feature $\frac{\log_2 N}{k}$
10	3.3	2.31 ± 0.14	1.4
26	4.7	3.26 ± 0.06	1.4
1708	10.7	5.58 ± 0.05	1.9

Table 1 above expresses results as bits of information per feature to identify the word.

In the case, with 10, 26, and 1708 words, the observer is using roughly 2, 3, and 6 features. The number of features needed to be detected to identify, k, grows logarithmically with the number of alternatives. For every twofold increase in vocabulary size, k grows some linear amount. It is an intuitive result.

Table 1 takes our results further. The information in bits required to identify a word is $\log_2 N$, where N is the number of possible words. 3.3 bits can distinguish 10 words, 4.7 bits can distinguish 26 words, and 10.7 bits can distinguish 1708 words. So we can assess the information provided by each feature by dividing the bits, $\log_2 N$ by the number of features, k. With our various alphabet sizes, the observers are consistently using about 1.7 bits/feature (range is 1.4 to 1.9). To provide 1.7 bits, a feature must have more than two values. If it has two values, present (0) or absent (1), the feature could provide only 1 bit. To provide 1.7 bits, the feature must take on $2^{1.7}$ = 3.2, or about 3, values.

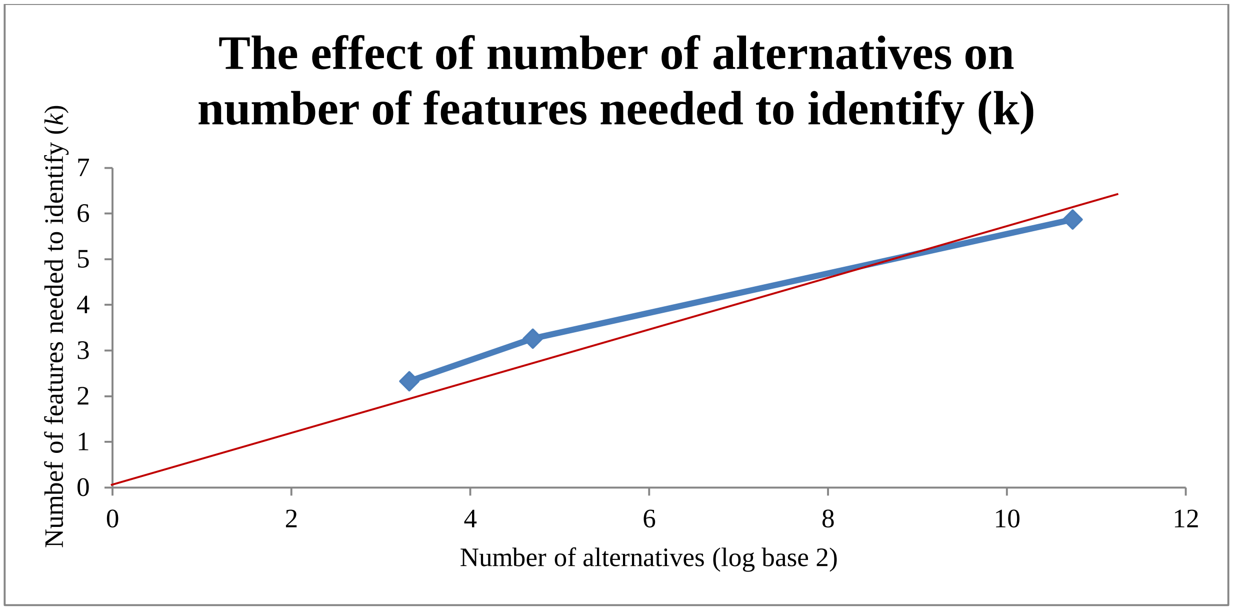


Figure 9: The graph above, with number of alternatives on a log base 2 scale and number of features needed to identify on a linear scale, shows that the *k* grows logarithmically with the number of alternatives. Taking the reciprocal slope of the red line (line of best fit), we end up with 1.83 bits of information for each feature.

In the context of words, then, each feature has 3 useful values that contribute to distinguishing among the possible words. Looking back at Admoni's experiment with gabor patches, each feature (gabor) had 2 useful values, either 0 or 1 change in orientation. When there were 2 gabors in a patch, there were 2^2 = 4 "letters", or alternatives. Then in a patch with 3 gabors there are 2^3 = 8 alternatives. In our sets of words, 2 features would distinguish $3^2 = 9$ words, 3 features would distinguish $3^3 = 27$ words, and 6 features would distinguish $3^6 = 729$ (just about half of 1708) words.

“Does word recognition involve preliminary letter identification?” asked McClleland and James (1976). In using textual stimuli, we must consider this. Letters themselves are complex objects; words are then a complex consolidation of them. And so we could employ nonword text strings that also do not appear to be words, a context for letters studied by Bjork and Estes (1973) in their study of linguistic masking, and how nonword performance stands to the test of alternatives. To further investigate how linguistic ability pervades word identification by manipulating how common the words are. Another possibility would be to employ non-word text strings that appear to the observer to have some linguistic properties (i.e. voller, quibbit, wug). For now, we have set up a basis for counting features in word identification.

Conclusion

We discovered that increasing the number of alternatives also increases the number of features needed to be detect in order to identify a word; the number of features needed grows logarithmically with the the vocabulary size. Furthermore, we found that each feature contributes 1.7 bits of information, corresponding to 3 values. Past work on detection of gabors assumed that each feature could contribute only one bit, corresponding to two possible values: present or absent. Our findings open up a new realm for understanding the visual hierarchy that extends from detecting a feature of an object to combining and making sense of the object. Our discovered effect of vocabulary size is an important limit to reading speed and comprehension.

Research on pattern detection has mostly studied very simple images, like gratings and plaids, and, partly as a consequence, there has been very little study with large numbers of alternatives. Words occur in a linguistic context, but we've treated them as pictures that convey information. Using them allowed us to easily work with thousands of alternatives. Words are arguably the most common way people absorb knowledge—people read to learn —so it is necessary to understand in detail how humans recognize them. We can relate our findings to linguistic phenomena such as priming (a function also of memory), where reading is made easier when the reader remembers which words are conceptually similar, thereby reducing alternatives for identifying the next word (Tabossi, Patrizia, and P. N. Johnson-Laird, 1980). Also, our discoveries may become helpful in optimizing font design. By better understanding visual processing, it becomes possible to project how humans can adapt to not only growing vocabulary size but also to the amount of objects to process in a world that only keeps developing.

Acknowledgements

I would like to amply thank the people who have guided me through this project and supported me in my first endeavor at laboratory research. Immeasurable gratitude towards Professor Denis Pelli, who built the foundation for this project and who has always lent an ear to my ideas. He has been a source of wisdom and inspiration throughout my time at his lab. Thank you to Jordan Suchow at Harvard University, who with Professor Pelli led me through the mathematics of the identification model based on feature detection, and who wrote the program that was critical to my data analysis. Interminable thanks to my research advisor Dr. Jonathan Gastel, whose knowledge knows few limits, for assisting me at every step of the way, from finding a mentor to pushing me to think critically from beginning to end. My deepest appreciation towards all the above for advising me through the revision process. Thanks to Vishal Shah, who has contributed greatly with his computer science expertise to the program on which the experiment ran and to formatting this paper. I would also like to thank the members of the Pelli lab at NYU’s Center for Neural Science for always providing helpful feedback and for making my first research experience a memorable one.

.

References

Admoni, H., Pelli, D.G. (2004) Counting Features: quantifying discrete parts in visual object identification. Unpublished manuscript.

Bjork, Estes (1973) Letter identification in relation to linguistic context and masking conditions* Memory & Cognition Vol. l,No. 3,217-223

Brainard, D. H. (1997). The Psychophysics toolbox. Spatial Vision, 10, 433–436.

Cobb, Percy W., and Frank K. Moss. "Four fundamental factors in vision."*Journal of the Franklin Institute* 205.2 (1928): 251-252.

Hubel, David H., and Torsten N. Wiesel. "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex." *The Journal of physiology*160.1 (1962): 106.

Kumar, A. (2012) "Palmprint Recognition Using Eigenpalms." Palmprint Recognition Using Eigenpalms. N.p., n.d. Web. 05 Nov. 2012. <http://www4.comp.polyu.edu.hk/~csajaykr/myhome/research/palmcode.html>.

Massaro, D. W., & Hary, J. M. (1986). Addressing issues in letter recognition. Psychological Research, 48(3), 123–32. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/3823333

McClelland, James L. (1976) Preliminary letter identification in the perception of words and nonwords. Journal of Experimental Psychology: Human Perception and Performance, Vol. 2 No. 1, 80-91.

Pelli, D. G. (1985) Uncertainty explains many aspects of visual contrast detection and discrimination. Journal of the Optical Society of America A 2, 1508-1532.

Pelli, D. G. (1997). The VideoToolbox software for visual psychophysics, transforming numbers into movies. Spatial Vision, 10, 437–442.

Pelli, Denis G., et al. "Feature detection and letter identification." *Vision research* 46.28 (2006): 4646-4674.

Robson, J. G., & Graham, N. (1981) Probability summation and regional variation in contrast sensitivity across the visual field. Vision Research Vol. 21, 409–418.

Tabossi, Patrizia, and P. N. Johnson-Laird. (1980) "Linguistic context and the priming of semantic information." *The Quarterly Journal of Experimental Psychology*32.4: 595-603.

Watson, A. B. (1979). Probability summation over time Vision Research 1979 Vol. 19, 515–522.