

To be submitted to Behaviour Research Methods

Running Header: ScanMatch

ScanMatch: A Novel Method for Comparing Fixation Sequences

Filipe Cristino¹, Sebastiaan Mathôt², Jan Theeuwes² & Iain D. Gilchrist¹

1 University of Bristol

2 Vrije Universiteit

Corresponding Author:

Iain D. Gilchrist

Department of Experimental Psychology,

University of Bristol,

12A Priory Road,

Bristol,

BS8 1TU

+44 (0)117 9289000

i.d.gilchrist@bristol.ac.uk

Abstract

We present a novel approach to compare saccadic eye movement sequences based on the Needleman-Wunsch algorithm used in bioinformatics to compare DNA sequences. In the proposed method, the saccade sequence is spatially and temporally binned and then recoded to create a sequence of letters which retains fixation location, time and order information. The comparison of two letter sequences is made by maximising the similarity score computed from a substitution matrix which provides the score for all letter pair substitutions and a penalty gap. The substitution matrix provides a meaningful link between each location coded by the individual letters. This link could be distance but could also encode any useful dimension including perceptual or semantic space. We show, by using synthetic and behavioural data, the benefits of this method over existing methods. The Scanmatch toolbox for Matlab is freely available online.

The last 30 years has seen an explosion of interest in eye movements as both a topic of study and a research tool to probe perceptual, cognitive and neural processes (Findlay & Gilchrist, 2003; Rayner, 2009). With an increase in the availability of eye trackers and a reduction in their cost, eye movement studies have not only flourished in vision research (e.g. Parkhurst, Law & Niebur, 2002; Tatler, Baddeley & Gilchrist, 2005) but also language (see Rayner, 1998, 2009), advertising (e.g. Pieters, Rosbergen, & Wedel, 1999; Maughan, Gutnikov, & Stevens, 2007), clinical research (e.g. Mosimann et al., 2005) and animal research (e.g. Kirchner & Thorpe, 2006). Although these fields differ widely in their aims and goals, one common denominator between them is the techniques used to study and analyse the eye movements.

The majority of studies in this area are concerned with the saccadic eye movement response. Saccadic eye movements are the fast ballistic movements that move the eye rapidly to point from one location to another. Between saccades the eye is stationary; these periods of time are referred to as fixations and it is during these time intervals that useful visual information is gathered (see Carpenter 1988 for more details).

Despite the growing interest in saccades and fixations, most methods of analysis rely on saccade latencies (time to initiate a saccade e.g. Zingale & Kowler, 1987), or the fixation locations as primary measures of behaviour. The vast majority of the literature on eye movements uses saccade amplitude or duration, number of fixations, fixation durations or other close derivatives as the main measures. However despite saccades being fundamentally sequential (one saccade is followed by the next one), very few methods are available to study their sequential properties. Markov models (Markov, 1971) have been used rather successfully in some domains such as eye movement modelling in reading (Engbert & Kliegl, 2001), but are complex to implement. Another, more widely applied method, which takes

into account the fixation order, is the Levenshtein distance (Levenshtein, 1966), more commonly referred to as the string edit method.

The Levenshtein distance measures the editing cost of transforming one string into another one using, in its basic form, a set of three operations (insertion, deletion and substitution) with a cost of one for each operation. The Levenshtein distance metric is given by the minimal score or minimum editing cost between two strings. More advanced versions can be found, either using the transposition as an extra operation (Wagner & Lowrance, 1975) or by assigning different weights to each operation (Okuda, Tanaka, & Kasai, 1976). Such methods have been successfully used by a number of authors to study saccade sequences (e.g. Noton & Stark, 1971; Brandt & Stark, 1997; Choi et al., 1995; Hacısalihzade, Allen, & Stark, 1992, Foulsham & Underwood, 2008). The method involves defining a number of spatial Regions of Interest (RoIs) in the scene being scanned and recoding the fixation sequence as a series of letters representing the fixated locations. Although the string edit method has proved to be a useful tool, and is relatively fast to compute, one of its main drawbacks is the lack of relationship between RoIs. As a result, the algorithm lacks flexibility as it cannot differentiate between close and distant regions or more generally between similar and dissimilar RoIs. A second drawback of this kind of method is that it does not take into account the fixation duration – all fixations however short or long are treated equally. It is clear that fixation duration is an important indicator of processing during a fixation (Henderson & Graham, 2008).

In this paper we describe a new method, used for nearly four decades in bioinformatics to analyse biological sequences (either DNA or protein sequences), to quantitatively score two eye movement sequences. The resemblance between both problems is striking; from the early days in gene sequence analysis it was discovered that sequences from related genes could be

classified as similar if a maximum of residues (string elements) matched along the sequence (Durbin, Eddy, Krogh, & Mitchison, 1998). This was a very important discovery since close similarity between two genes or proteins is a strong argument for their homology (common ancestry). To solve the string matching problem, collaboration between computer scientists, mathematicians and biologists resulted in a large number of optimised methods called sequence alignment algorithms. These methods are optimised to the extreme because in this field it is often necessary to compare millions of sequences together (i.e. in the Genome project). The methods developed in this area range from robust and accurate but slow such as the Needleman-Wunsch algorithm (Needleman & Wunsch, 1970), to very fast but without the guarantee of finding the optimal solution such as the BLAST algorithm (Altschul et al., 1997). In this paper we show how the methodology of sequence alignment (Needleman-Wunsch algorithm) can be applied to eye movements, and then present three experiments in which the method (which we have called ScanMatch) is used. The ScanMatch application was coded in Matlab with the bioinformatics toolbox and the code is freely available to download online or can be requested directly from the authors.

Methods

Creating a sequence

Eye movement data collected through any eye tracker can be used to create a sequence of eye movements to which the ScanMatch method can be applied. First, the data needs to be filtered down to saccades and fixations. Then, to be able to code string sequences, images have to be divided into Region of Interests (RoIs), which can either be designated feature regions within an image (for example eyes, mouth, nose in a face image; or doors, windows and people in an interior scene), or can be created by simply binning the image into a discrete

number of regular bins. A letter is then assigned to each region and every eye fixation within that region is tagged with its name. One problem with previous string based methods was the lack of coding of fixation duration. Here to take fixation time into account, we introduce temporal binning into the string by repeating the letter corresponding to the region of interest in a way that is proportional to the fixation duration. In this way, the string created by the saccade sequence incorporates spatial location, sequential information and temporal durations (see Figure 1). In Experiments 2 and 3, we set this temporal sampling to 50ms as this value allows an accurate sampling of the large variability of fixation times (typically 100 to 1000ms).

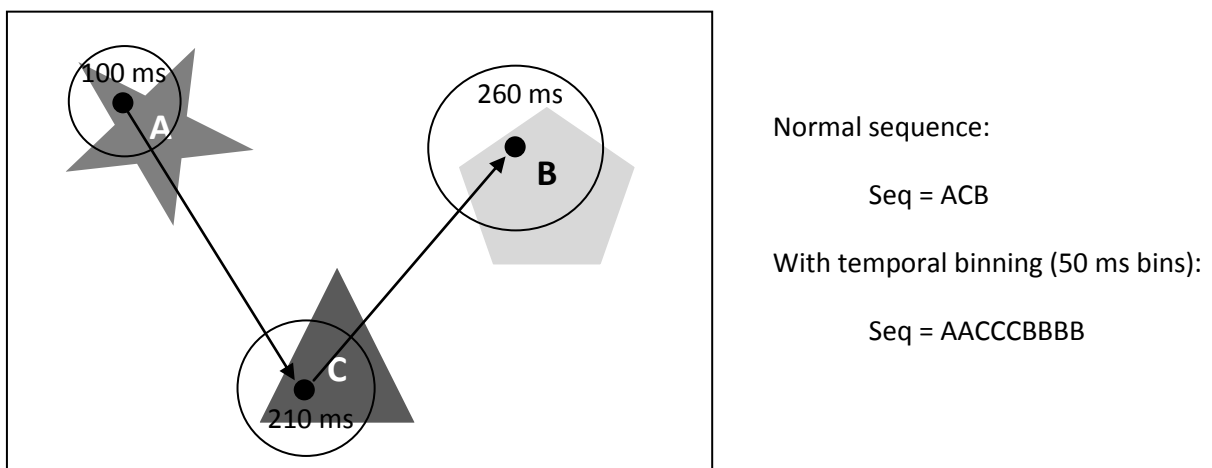


Figure 1: From fixations to eye movement sequences; each fixation is given the letter of the Region of Interest where it landed (in this example ACB). To take into account fixation durations, dwell times can be sampled (in this example with 50 ms bins) with each sample given the RoI letter of the fixation resulting in the following sequence: AACCCBBBB.

Another practical factor which can become an issue when using strings to represent eye movement sequences is that, in some circumstances, one may need more than 26 RoIs (the length of the English alphabet). To avoid this, a double string procedure was implemented where each region is coded by two letters; for ease of reading the first letter will be in lower

case and the second in upper case. The second letter does not need to go up Z. If, for example, an image is divided in 12 by 2 bins, one may choose to name the first row bins aA to aL, and the second bA to bL (hence using a modulus of 12). As part of the ScanMatch software a Matlab procedure is provided to create double strings from RoIs numbers which takes into account the modulus.

Comparing strings

The aim of the method presented in this paper is to quantitatively describe how similar two eye movement sequences are (i.e.: “aAaBaCaD” with “aAaCaCaD” or “aEaAaTaS”). For this, we used a family of algorithms widely used in bioinformatics called sequence alignment algorithms.

Sequence alignment

Global sequence alignment algorithms seek the optimal alignment over the entire length of two sequences by maximising its score. The sequences are aligned according to a substitution matrix which gives scores for every alignment (i.e. aligning aA with aB may give a score of 5, whereas aligning aA with aK gives a score of 2; and aligning aA with aA may give the maximum score of 10). To be able to perform the sequence alignment, gaps (spaces between two residues) can be introduced in either sequence at a certain cost (i.e. -1). Gap to gap alignment is not allowed as they are redundant. The implementation of such algorithm has been extensively researched in the computational biology field, with one of the most popular implementations being BLAST (Altschul et al., 1997), the method used for the Human Genome project. The implementation of this method has been very successful, but it prioritises speed over accuracy because it is intended to be used for analysis of enormous

datasets (millions of entries). For the purpose of this paper, we chose the Needleman-Wunsch algorithm; a slower but far simpler method, which is guaranteed to find the optimal solution.

Needleman-Wunsch algorithm

The Needleman-Wunsch algorithm uses dynamic programming to solve the alignment problem. The basic idea is that it uses local optimal alignment of sub-sequences to create the best overall alignment. By doing this, it reduces the number of possibilities to be considered but still guarantees the optimal solution. The algorithm is divided in two parts; the first creates a matrix with all scoring possibilities based on a substitution matrix and a gap penalty, and the second seeks the optimal alignment, tracing back the matrix from the top left corner to the outmost column or row by selecting the optimal route (route adding up to the maximum score). The score is given by the optimal route throughout the matrix; the higher the score, the more likely the two sequences are similar. In this algorithm only two parameters need to be set; the substitution matrix and the gap penalty.

The substitution matrix

The substitution matrix provides the algorithm with a score (the higher the better) for aligning two letters. As previously discussed, one of the main drawbacks of the string edit method is the lack of relationship between each RoI. In the current method, this problem is addressed with the substitution matrix as it can encode information about the relation between each RoI. In bioinformatics several matrices are available with the most popular being PAM_{xx} and BLOSUM_{xx} (Mount, 2008). PAM_{xx} provides scores based on the observed frequencies of alignments in related proteins (xx meaning up to xx% of divergence between two genes i.e. xx=50), where identities are given the highest scores (frequently observed substitutions are given a positive score and rarely observed substitutions a negative score). Using similar logic,

for sequences of fixations, within the substitution matrix, the relationships between RoIs in an image can be coded in a number of ways. A basic relationship is the one based on distance with scores in the substitution matrix inversely related to the Euclidian distance between each bin, with the highest score given to the identities and the lowest to the bins the furthest apart. However, the relationship does not have to be spatial; for example, a relationship can be based on colour, (e.g. in a visual search task), or a more semantic segmentation of an image (e.g. clothes or toys in some interior scene). One important parameter when constructing the substitution matrix is the cut-off point of what should be positive (highly related items) and what should be negative (loosely related items). In the case of eye movements, where the substitution matrix is based on distance between RoIs, one could use the variability in saccade landing to decide the cut-off point for which values in the substitution matrix should become negative (threshold). Using this method, for the distance based substitution matrix, it means that the alignment algorithm will aim to align only regions which are within the variability of saccade landing. We therefore define this threshold to two standard deviation of all the saccade amplitude of a particular experiment from which the sequences are compared. A Graphical User Interface (GUI) to construct and visualise substitution matrix based on distance is provided with the Scanmatch toolbox.

The gap penalty

The gap penalty is the second and last free parameter for the string alignment algorithm and is directly related to the substitution matrix. Its value is the score of aligning any element in a sequence with a gap (a free space). A large gap penalty will favour the insertion of many gaps and will therefore discourage the alignment of less highly related residues, whereas a small or negative gap penalty will favour the alignment of residues (even if only loosely related) and discourage gaps. One could see the gap penalty as the limit for which two residues should not

be aligned, but instead, a gap created. Figure 2 describes a simple example using two different values for the gap penalty.

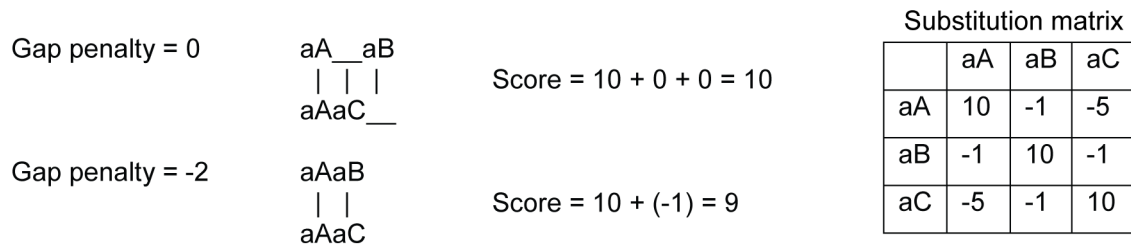


Figure 2: Simple case of comparing the following two short sequences with on the right an example of a substitution matrix to demonstrate the effect of the gap penalty.

As it can be seen in Figure 2, the gap penalty acts as a similarity threshold, below which the insertion of a gap is favoured over alignment. In longer sequences, gaps may be inserted if this benefits the global alignment of the sequences, even though this may result in suboptimal local alignments. If the threshold in the substitution matrix is well chosen, as previously explained, the gap value can be set to zero. In the three experiments described in this paper, the gap value is always set to zero.

The scoring

The alignment score given by the Needleman-Wunsch algorithm is highly dependent on the substitution matrix and on the length of the sequences compared. Indeed, comparing two identical strings with 20 residues will naturally give a higher score than two identical sequences with two residues. To overcome these issues, a normalisation of the score is performed:

$$Normalised\ score = \frac{score}{Max(substituion\ matrix) * length\ of\ the\ longest\ sequence}$$

Using this normalisation the best possible match between two sequences will give a score of 1. Examples of sequence comparisons can be found online on the Scanmatch webpage as part of a ScanMatch toolbox tutorial.

Matlab code

ScanMatch is available for download from <http://www.scanmatch.co.uk> or can be requested directly from the authors. It provides a set of Matlab script files which allow the comparison of the sequences. The Needleman-Wunsch algorithm implementation under Matlab was taken from the bioinformatics toolbox. Although the ScanMatch program does not need the bioinformatics toolbox to run, a licence for this toolbox should be held. A user interface (Figure 3) is provided to facilitate the visualisation of the algorithm. The Matlab files are annotated and a tutorial on how to use the toolbox can be found on the download webpage.

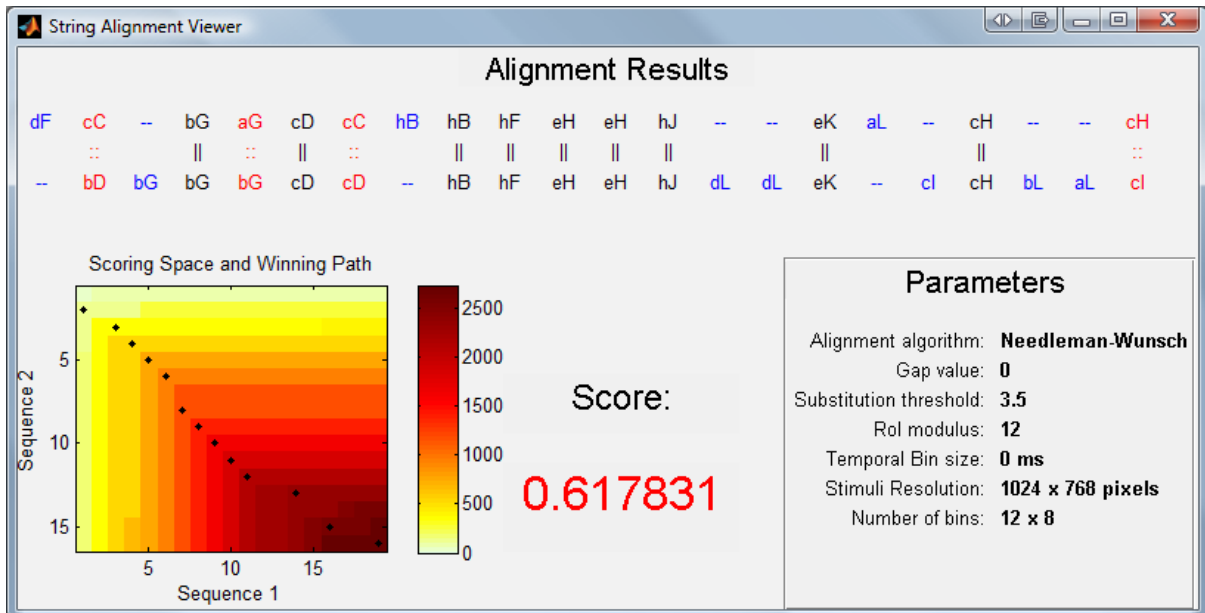


Figure 3: The ScanMatch Gui (V1.00) under Matlab. It gives a visual representation of the string alignment method.

To test our new ScanMatch method we used three datasets; a synthetic one which, by varying a set of parameters, enabled us to observe how well the algorithm behaved, and two data sets taken from simple eye tracking experiments.

Experiment 1

This first experiment was based on synthetic data to demonstrate the power of the proposed approach in describing eye movement sequences, and to compare our results against the well established Levenshtein measure (string edit distance).

Method and Results

We simulated a simple experiment in which saccades were generated between 26 linearly arranged tiles. The starting point for scanning was either from the left (from tile aG to tile aT) or from the right (tile aT to tile aG) as described in Figure 4. We predicted that this would result in two different types of eye movement sequences, varying only temporally (order of the visited tiles), but with the same spatial statistics (with each tile being visited the same number of times in whichever direction the task was performed).

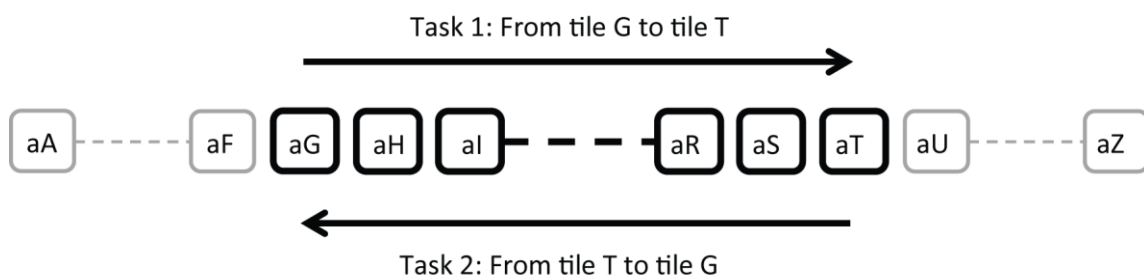


Figure 4: Experiment 1; stimuli and task. 26 tiles are arranged linearly (from aA to aZ) where the two simulated tasks were to go from tile aG to aT (for task one) or go from tile aT to tile aG (for task two) fixating each tile in-between.

String sequences were created as described in Figure 5 where, by varying a single parameter (σ), we were able to create sets of sequences which varied in similarity. For this simple

simulation we assumed constant fixation time, hence each simulated fixation is represented by a single tile.

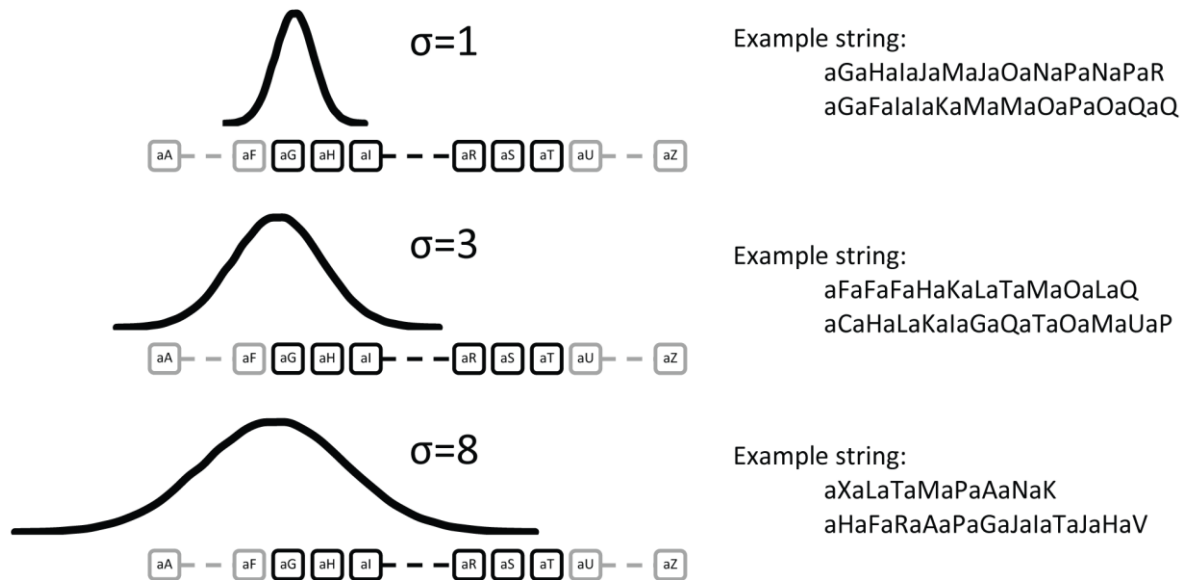


Figure 5: Experiment 1; sequence creation. As the standard deviation of the polling distribution increases, the variability in landing increases making the generated sequences more and more different. If a saccade lands outside the 26 tiles (smaller than aA or greater than aZ) this sample is deleted and not replaced (hence the likelihood of the sequences getting shorter as the noise increase).

Strings are created by sampling a normal distribution centred on each tile to be visited (tile aG to aT) and with a standard deviation σ . If a sample is outside the range of tiles (lower than aA or greater than aZ) it will be deleted and not replaced. In such a case the sequence will have fewer letters. If the standard deviation of the distribution (σ) is equal to zero then each string will be 12 letters long in a perfect order. As σ increases, the strings will diverge more and more, and there will be more variability in the length of the sequences (Figure 6a). In this experiment, a total of 24 sets of strings for each σ level were created, each set containing 100 sequences for the first task (going from tile aG to aT) and 100 for the second (going from tile aT to aG). We chose these numbers to match the numbers of trials in a typical experiment.

The parameter σ was varied from 0 to 14 by steps of 0.5. Figure 6a shows the mean string length of all generated strings across the different σ levels (error bars represent 95% Confidence Intervals), while Figure 6b shows the mean distance between two consecutive string residues (equivalent to saccade amplitude in a real experiment) in generated strings. The bigger the spread of the distribution is, the greater the variation of the strings in content and in size.

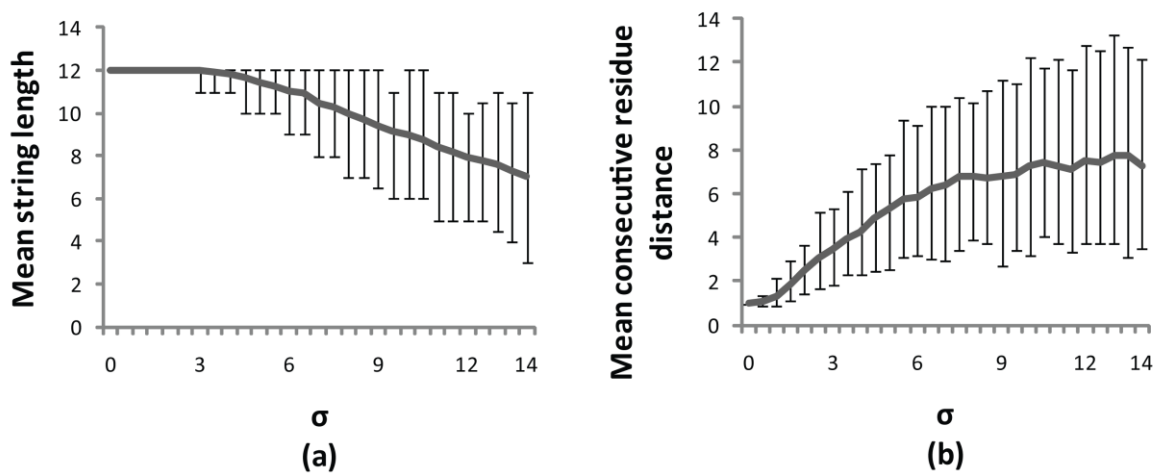


Figure 6: Effect of σ on the generated sequences. (a) As the standard deviation of the distribution increased the mean string length of a sequence decreased allowing different length sequence comparisons (error bars represent 95% confidence intervals). (b) The distance between two consecutive residues increased as the σ gets higher (i.e aA to aB have a distance of one while aA to aD have a distance of 4).

The substitution matrix used here was based on distance as outlined in the previous session and created with a standard deviation of three. This value is the mean standard deviation of the distance between two consecutive residues across all noise conditions. The same substitution matrix was used to compare all strings across all the levels. In this experiment, the sequences were compared using our new ScanMatch method as well as using the classic

Levenshtein distance method (a single character string was used as the number of RoIs was under 26).

Sequences of each set (task one or two) were compared with each other, i.e. strings from the first set were compared to all other strings in that set, and also to strings in the second set. It follows, therefore, that comparisons within sets should lead to higher scores, whereas comparisons across sets should lead to lower scores. A k-mean clustering algorithm was then used to classify each comparison set to either the first set (a within set comparison) or second set (between set comparison). The data was analysed by measuring the percentage of incorrect classifications (with chance giving a misclassification of 50%). Figure 7 shows the mean percentage of misclassification for the 24 sets, where the error bars represent the 95% confidence intervals.

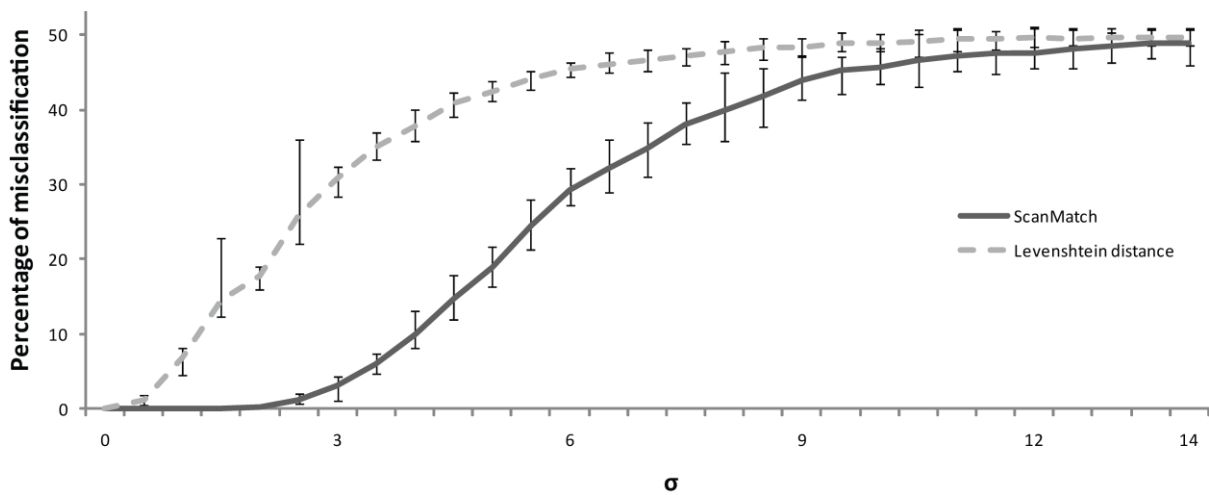


Figure 7: Experiment 1 results. A k-mean clustering algorithm was used to classify the sequences per task. ScanMatch outperform the Levenshtein distance up to a very high level of noise (8) with no misclassifications up to a σ level of 2.5.

Our new ScanMatch method performs very well even with noisy data, only performing at chance when σ is greater than 10. It successfully classifies sequences without a single error up to a noise level of 2, whereas the Levenshtein distance method starts to misclassify as soon as noise is added ($\sigma > 0.5$). At this level of noise, the string edit method already misclassified 18% of the trials as the sequences can be very different with that amount of noise.

ScanMatch performs significantly better than the Levenshtein distance method up to an added noise level with a standard deviation of 8.5 (non-overlapping error bars in Figure 7), at which point the string edit method is already performing at chance. This experiment shows how well our new method behaves even with a higher level of divergence in the input sequences.

On a modern personal computer system (Intel Quad core clocked at 3.0Ghz with 4GB of Ram running under Vista x64 with Matlab 2009a on a single core) the ScanMatch method took less than two minutes to perform the 100,000 sequence comparison needed to compare a block of 100 sequences.

Experiment 2

In this experiment we assessed the performance of the new method using a simple sequential looking task with human eye movement data.

Methods and Results

The stimuli, displayed on a computer screen, were red and green numbers ranging from 0 to 9 (Figure 8). A single participant (one of the authors - SM) was asked on a given trial to saccade from number to number in ascending (1 to 9) or descending (9 to 1) order and to follow either the green or red numbers while their eye movements were recorded. Instructions were given to the participant before each trial, explaining which colour and which order to

follow. In a single block, the participant performed the 20 trials in random order (4 conditions performed 5 times).



Figure 8: Example stimuli for Experiment 2. The participant was asked to fixated either all the blue or red numbers in a ascending or descending order

Eye movements were recorded using an EyeLink II, produced by SR Research. The EyeLink II is a head-mounted binocular eye-tracking device with a sampling rate of 500 Hz. The stimuli were displayed on a CRT screen with a screen resolution set to 1024 by 768 pixels. A nine point calibration was performed before the start of the experiment and a drift correction before each trial. Saccade extraction was done using SR Research normal saccadic filter with values of 30 deg.s^{-1} velocity threshold and 8000 deg.s^{-2} acceleration threshold. The participant used a chin-rest situated 66 cm from the display screen.

In this experiment, as the stimuli were constant across task-instruction, the eye movements were binned spatially with a grid size of 12 by 8 with a label starting from aA to aL on the first line to hA to hL on the last. Each eye movement in a bin was given its label, creating a total of 20 sequences. The substitution matrix was based on the distance between each bin (as describe above with a cut-off value of 5.5) and with a gap value of 0.

Each sequence was compared to every other sequence using our ScanMatch method, creating a 20 by 20 score matrix. The eye movement sequences of a participant performing the same condition should be more similar than when performing a different condition. To check this hypothesis a k-means algorithm was used to cluster the scoring matrix for each comparison set (a single trial against all of the others). The self comparison (leading to a perfect score of 1) was removed from the clustering process. The mean number of saccades per trial was 18.55 (M=18.55, SD=2.18) with a mean amplitude of 6.5 degrees (M=6.5, SD=11.5). Note that this SD value is the one used to construct the substitution matrix.

Comparisons of same condition trials scored a mean normalised score of 0.72 (M=0.74, SD=0.034), while comparisons of different condition trials scored mean normalised score of 0.15 (M=0.15, SD=0.01), a significantly lower score ($F(1, 32) = 7592, p < 0.001$). Trials were perfectly clustered with not a single error. Even though the spatial statistics between colour conditions were the same (the participant had to fixate each number the same number of times in order to execute the task), using the temporal aspect of the sequences, ScanMatch clustered all the trials successfully. This experiment shows the effectiveness of our new method when used with RoIs created by binning the image and a substitution matrix purely based on distance.

Experiment 3

In this experiment we collected human eye movement data from a number of participants on a visual search task and investigate if the new method is able to differentiate between conditions.

Method and Results

In this visual search task participants had to report the orientation of a letter T, tilted to 90 degrees, while their eye movements were recorded. As soon as they located the target, participants were asked to press either the left or right arrow keys to report the target's orientation.

The randomly generated stimuli consisted of a target and 19 distractors. These were a combination of T and Ls in various orientations, 10 of which were a different colour from the target (red or green) and 9 of which were the same colour (Figure 9 shows an example display with a red target). The letters had a size of 0.375 visual degrees and were separated by a gap of at least 2.8 degrees. The monitor was calibrated to display the red and green letters with the same luminance. Participants knew the colour of the target before each trial and the same stimuli was presented across participants but in a different order. Eye movement recording methods and display characteristics were the same as in Experiment 2.

Eight participants took part in this experiment, performing 200 trials each (100 searching for a red target and 100 for a green target). All had normal or corrected to normal vision.

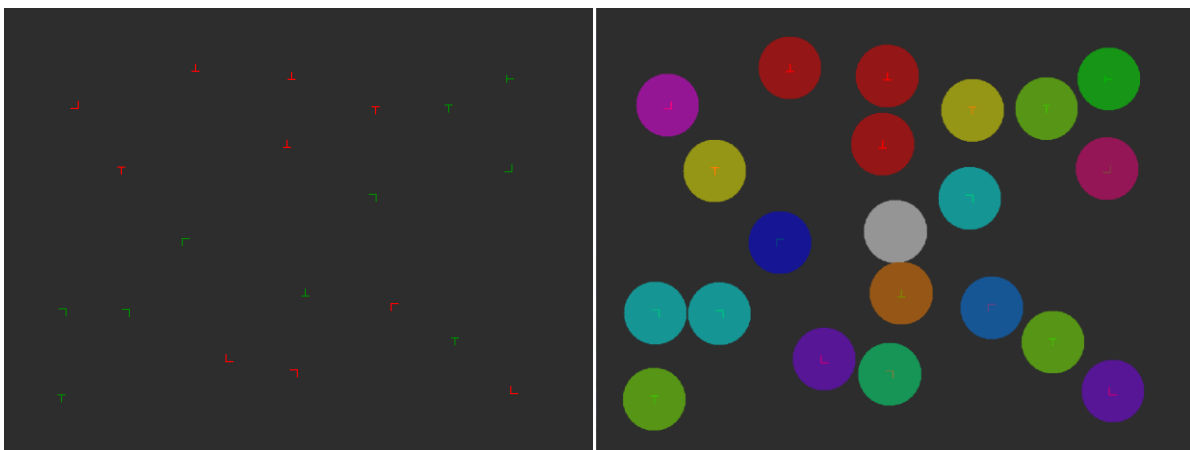


Figure 9: Example display (left hand panel) and corresponding regions of interest (right hand panel) in Experiment 3. Each type of distractors and targets were assigned a ROI name.

As the spatial location of the target and distractors were different from trial to trial, the regions of interest were based on the colour and orientation of the T and Ls. The ROIs created were circular, centred on the middle of the letter with a diameter of 80 pixels (2.8 degrees) as represented in Figure 9. In this experiment, to demonstrate the flexibility of our algorithm, we used a substitution matrix in which we coded colour as our main measure. We therefore coded a substitution of a green object with a green object a high value (in the case of searching for a green target), and a substitution to a red object as a low value (any substitution with the background or starting point was coded to 1).

In this experiment, the gap value was left at zero. The analysis was carried out on a per subject basis, where every eye movement sequence within the subject was compared to every other sequence within and across task (the hypothesis being that eye movement sequences when searching for a target of the same colour will be more similar than those searching for a target of a different colour). We therefore ran the ScanMatch method twice; once with the green substitution matrix where compared trials searching for the green target should score highly, and once with the red substitution matrix where compared trials searching for the red target should score highly.

Using our ScanMatch method, trials compared within each task have a significantly higher normalised score (red trials: $M=0.68$, $SD=0.02$, green trials: $M=0.66$, $SD=0.013$) than trials compared across tasks (red trials: $M=0.26$, $SD=0.013$, green trials: $M=0.25$, $SD=0.014$) and this difference was highly reliable ($F(1, 28) = 5793$, $p < 0.001$). The clustering was very successful with 95% ($M=0.95$, $SD=0.03$) of red and 94% ($M=0.94$, $SD=0.05$) of green trials correctly clustered. This experiment demonstrates the success and potential of a substitution matrix in which a non-spatial (in this case perceptual) dimension can be coded.

Discussion

We present a new method – ScanMatch - for comparing sequences of fixation generated by eye movements. The method addresses two significant limitations of the previously used string edit methods. First, the method allows the durations of the fixations to be taken into account and included in any assessment of the similarity between two fixation sequences. Second the method allows identified regions of interest to be classified. At it's most simple this means the spatial distance between regions of interest (RoIs) can be taken into account such that RoIs that are close to each other can be considered 'less different' than RoIs that are further apart. Importantly, this method allows RoIs to be classified as more similar, or different, on any dimension be it cognitive, perceptual or task based.

We present three experiments to test the performance of these methods as applied to fixation sequences. In the first experiment, in which we use synthetic data we make a direct comparison between string edit methods and our proposed new method. This is possible here because we assume that all fixation durations are the same and it is only under these circumstances that we can compare the two methods directly. The ScanMatch method reliably outperforms the string edit method at identifying the original simulated groups and this is particularly true as more noise is introduced in to the groups to make the task of differentiating between the groups harder.

In the subsequent experiments we showed the ability of the method to deal with real eye movement data and differentiate between conditions in a robust manner. The method was stable and reliable both for a small data set from a single participant (Experiment 2) and for data sets from multiple participants of the size that occurs in a typical experiment (Experiment 3). We demonstrated that the methods could code similarity between regions in

spatial coordinates (Experiment 2) as well as in perceptual coordinates - in this case colour (Experiment 3).

At the heart of this new method is the Needleman-Wunsch algorithm which has been developed to be computationally efficient without compromising on the quality of the fit between strings. As a result the comparison of large numbers of strings is possible on a modern desktop computer.

The sequence of fixations is a complex data structure and care needs to be taken with the application of this and other methods to be explicit about the assumption that underlie the similarity calculation been computed. One of the advantages of the current method is that these assumptions are explicitly held within a single matrix (the substitution matrix).

However this matrix is simple in that it encodes a single substitution value between two RoIs. The assumption here is that this value is stable across the experiment and therefore will not, for example, change over time. This may be a particularly important consideration when using these methods for experiments that involve learning.

These methods are particularly strong when the ordering of the fixations may be an important determinate of a change in the behaviour of the participant. However if the sequence in which the information is encoded is not important, and so different fixation orders will not have an effect on cognition or perception these methods may not be as useful.

In conclusion, our new method, ScanMatch, can differentiate between conditions and is robust to the substantial noise inherent in fixation sequences. As a method it opens up the possibility of studying in far more detail, and with more precision, how eye movements influence performance in complex tasks where a number of eye movements are generated in a sequence.

Acknowledgement

This work was supported by the NWO & ESRC Bilateral Grant Scheme.

References

- Altschul, S., Madden, T., Schaffer, A., Zhang, J., Zhang, Z., Miller, W., et al. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, 25(17), 3389-3402.
- Brandt, S. A., & Stark, L. W. (1997). Spontaneous eye movements during visual imagery reflect the content of the visual scene. *Journal of Cognitive Neuroscience*, 9, 27–38.
- Buswell, G. (1935). *How people look at pictures a study of the psychology of perception in art*. Chicago Ill.: The University of Chicago press.
- Carpenter, R. H. S. (1988). *Movements of the Eyes. (2nd edition)*. London: Pion.
- Choi, Y. S., Mosley, A. D., & Stark, L. W. (1995). String editing analysis of human visual search. *Optometry and Vision Science*, 72, 439–451.
- Durbin, R., Eddy, S. R., Krogh, A., & Mitchison, G. (1998). *Biological sequence analysis* (p. 356). Cambridge University Press.
- Engbert, R., & Kliegl, R. (2001). Mathematical models of eye movements in reading: a possible role for autonomous saccades. *Biological Cybernetics*, 85(2), 77-87.
- Findlay, J. M. & Gilchrist I. D. (2003). *Active vision: The psychology of looking and seeing*. Oxford University Press, Oxford.
- Foulsham, T., & Underwood, G. (2008). What can saliency models predict about eye movements? Spatial and sequential aspects of fixations during encoding and recognition. *Journal of Vision*, 8, 1–17.

- Hacisalihzade, S. S., Allen, J. S., & Stark, L. (1992). Visual perception and sequences of eye movement fixations: A stochastic modelling approach. *IEEE Transactions on Systems, Man and Cybernetics*, 22,474–481.
- Henderson, J. M., & Graham, P. L. (2008). Eye movements during scene viewing: evidence for mixed control of fixation durations. *Psychonomic Bulletin & Review*, 15(3), 566-73.
- Itti, L., & Koch, C. (2001). Computational modelling of visual attention. *Nature Reviews Neuroscience*, 2(3), 194-203.
- Kirchner, H., & Thorpe, S. J. (2006). Ultra-rapid object detection with saccadic eye movements: Visual processing speed revisited. *Vision Research*, 46(11), 1762-1776.
- Levenshtein, V. (1966). *Binary codes capable of correcting deletions, insertions, and reversals* (pp. 710, 707).
- Markov, A. (1971). Extension of the Limit Theorems of Probability Theory to a Sum of Variables Connected in a Chain. In *Dynamic Probabilistic Systems (Volume I: Markov Models)* (pp. 577, 552). John Wiley & Sons, Inc.
- Maughan, L., Gutnikov, S., & Stevens, R. (2007). Like more, look more. Look more, like more: The evidence from eye-tracking. *Journal of Brand Management*, 14(4), 335-342.
- Mosimann, U. P., Muri, R. M., Burn, D. J., Felblinger, J., O'Brien, J. T., & McKeith, I. G. (2005). Saccadic eye movement changes in Parkinson's disease dementia and dementia with Lewy bodies. *Brain*, 128(6), 1267-1276.

- Mount, D. W. (2008). Comparison of the PAM and BLOSUM Amino Acid Substitution Matrices. *Cold Spring Harbor Protocols*, 2008(7).
- Needleman, S. B., & Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3), 443-453.
- Noton, D., & Stark, L. (1971). Scanpaths in Eye Movements during Pattern Perception. *Science*, 171(3968), 308-311.
- Okuda, T., Tanaka, E., & Kasai, T. (1976). A Method for the Correction of Garbled Words Based on the Levenshtein Metric. *IEEE Transactions on Computers*, C-25(2), 172-178.
- Parkhurst, D., Law, K., & Niebur, E. (2002). Modeling the role of salience in the allocation of overt visual attention. *Vision Research*, 42(1), 107-123.
- Pieters, R., Rosbergen, E., & Wedel, M. (1999). Visual Attention to Repeated Print Advertising: A Test of Scanpath Theory. *Journal of Marketing Research*, 36(4), 424-438.
- Rayner, K. (2009). Eye movements and attention in reading, scene perception and visual search. *The Quarterly Journal of Experimental Psychology*, 62, 1457-1506.
- Rayner, K. (1998). Eye Movements in Reading and Information Processing: 20 Years of Research. *Psychological Bulletin*, 124, 372-422.
- Tatler, B. W., Baddeley, R. J., & Gilchrist, I. D. (2005). Visual correlates of fixation selection: effects of scale and time. *Vision Research*, 45(5), 643-659.

Wagner, R. A., & Lowrance, R. (1975). An Extension of the String-to-String Correction Problem. *J. ACM*, 22(2), 177-183.

Yarbus, A. L. (1967). *Eye Movements and Vision*. Plenum Press.

Zingale, C. M., & Kowler, E. (1987). Planning sequences of saccades. *Vision Research*, 27(8), 1327-1341.