Identifying Fixations and Saccades in Eye-Tracking Protocols

Dario D. Salvucci
Nissan Cambridge Basic Research
Four Cambridge Center
Cambridge, MA 02142 USA
+1 617 374 9669
dario@cbr.com

Joseph H. Goldberg
Dept. of Industrial and Manufacturing Engineering
Pennsylvania State University
310 Leonhard Building
University Park, PA 16802 USA
+1 814 863 2370
jgoldberg@psu.edu

ABSTRACT

The process of fixation identification—separating and labeling fixations and saccades in eye-tracking protocols—is an essential part of eye-movement data analysis and can have a dramatic impact on higher-level analyses. However, algorithms for performing fixation identification are often described informally and rarely compared in a meaningful way. In this paper we propose a taxonomy of fixation identification algorithms that classifies algorithms in terms of how they utilize spatial and temporal information in eye-tracking protocols. Using this taxonomy, we describe five algorithms that are representative of different classes in the taxonomy and are based on commonly employed techniques. We then evaluate and compare these algorithms with respect to a number of qualitative characteristics. The results of these comparisons offer interesting implications for the use of the various algorithms in future work.

Keywords

Fixation identification, eye tracking, data analysis algorithms.

1. INTRODUCTION

Eye tracking has been gaining in popularity over the past decade as a window into observers' visual and cognitive processes. For instance, researchers have utilized eye tracking to study behavior in such domains as image scanning [e.g., 12], driving [e.g., 11], arithmetic [e.g., 20], analogy [e.g., 17], and reading [e.g., 14]. In these and other domains, researchers typically analyze eye movements in terms of fixations (pauses over informative regions of interest) and saccades (rapid movements between fixations). Common analysis metrics include fixation or gaze durations, saccadic velocities, saccadic amplitudes, and various transition-based parameters between fixations and/or regions of interest.

The analysis of fixations and saccades requires some form of *fixation identification* (or simply *identification*)—that is, the translation from raw eye-movement data points to fixation locations (and implicitly the saccades between them) on the visual display. Fixation identification significantly reduces the size and complexity of the eye-movement protocol, removing raw saccade data points and collapsing raw fixation points into a single representative tuple. This reduction is useful for at least two reasons. First, little or no visual processing can be achieved during a saccade [6], and thus the actual paths traveled during saccades are typically irrelevant for many research applications. Second, smaller eye movements that occur during fixations, such as tremors, drifts, and flicks [1, 4], often mean little in higher-level analyses. Thus, fixation identification is a convenient method of minimizing the complexity of eye-tracking data while retaining its most essential characteristics for the purposes of understanding cognitive and visual processing behavior.

Fixation identification is an inherently statistical description of observed eye movement behaviors. While it is generally agreed upon that visual and cognitive processing do occur during fixations [e.g., 9], it is less clear exactly when fixations start and when they end. Thus, regardless of the precision and flexibility associated with identification algorithms, the identification problem is still a subjective process. Therefore one efficient way to validate these algorithms is to compare resultant fixations to an observer's subjective impressions.

Though widely employed, fixation identification and its various implementations have often been given short shrift in the eye-movement literature, particularly in literature concerning the interaction of eye movements and higher-level cognition. However, identification is often a critical aspect of eye-movement data analysis that can have significant effects on later analyses. For instance, poorly defined identification algorithms may produce too many or too few fixations, or may be overly sensitive to outlier data points, thus biasing interpretation. Karsh and Breitenbach [10] provided a rigorous demonstration of how different identification algorithms can produce vastly different interpretations even when analyzing identical protocols. Thus, good identification algorithms ensure valid fixation and saccade locations and durations which in turn influence inferences such as processing complexity and visual search paths.

In addition to the frequent underspecification of identification algorithms, little work has been done in evaluating and comparing different possible algorithms. Researchers and practitioners often have minimal information to guide their decision of which algorithm to use in particular situations. Not surprisingly, this problem leads to the somewhat haphazard application of different algorithms, making it difficult to compare results derived by different methods of identification.

In this paper we address these problems by proposing a novel taxonomy of fixation identification algorithms and evaluating representative algorithms in the context of this taxonomy. The taxonomy classifies algorithms with respect to five spatial and temporal criteria. The spatial criteria divide algorithms in terms of their use of velocity, dispersion, and area-of-interest information. The temporal criteria divide algorithms in terms of their use of duration information and their local adaptivity. The taxonomy provides a meaningful labeling and classification of existing algorithms so that they may be more easily compared in a systematic way to guide the choice of algorithms for particular applications.

To demonstrate the usefulness of the taxonomy, we identify and describe five algorithms that are representative of different classes in the taxonomy. We first provide rigorous descriptions of these algorithms in "pseudocode" form to best illustrate the similarities and differences between the algorithms. We then evaluate and compare the methods with respect to several qualitative characteristics including algorithm speed, accuracy, robustness, parameter space, and ease of implementation. We conclude with a discussion of implications and recommendations for how the various methods can be best utilized for future applications.

2. A TAXONOMY OF FIXATION IDENTIFICATION ALGORITHMS

Our taxonomy classifies fixation identification algorithms with respect to spatial and temporal characteristics, as summarized in Table 1. In constructing this taxonomy, we attempted to identify a minimal set of criteria that would best capture the differences between common existing algorithms. While we could propose a more complex taxonomy that could potentially account for finer distinctions and/or hybrid algorithms, the proposed basic taxonomy provides a useful characterization of the *primary* types of identification algorithms. The taxonomy will thus serve as a good starting point for comparing and evaluating existing identification algorithms, as we discuss in the following two sections.

For spatial characteristics, we identify three criteria that distinguish three primary types of algorithms: velocity-based, dispersion-based, and area-based. Velocity-based algorithms emphasize the velocity information in the eye-tracking protocols, taking advantage of the fact that fixation points have low velocities and saccade points have high velocities. (Note that, assuming a constant sampling rate, velocities are simply distances between sampled points and thus we can ignore the temporal component implicit in velocities.) Dispersion-based algorithms emphasize the dispersion (i.e., spread distance) of fixation points, under the assumption that

Table 1: Taxonomy of fixation identification algorithms, sample citations for their use, and the sample algorithms presented here.

Criteria		Representative Algorithms				
		I-VT	I-HMM	I-DT	I-MST	I-AOI
Spatial	Velocity-based	X	X			
	Dispersion-based			X	X	
	Area-based					X
Temporal	Duration sensitive			X		X
	Locally adaptive		X	X	X	

fixation points generally occur near one another. Area-based algorithms identify points within given areas of interest (AOIs) that represent relevant visual targets. These algorithms, unlike the others described here, provide both lower-level identification and higher-level assignment of fixations to AOIs [see 15, 16]. Because fixations can also be used as inputs to AOI algorithms, these can also represent higher levels of attentional focus on a display. These dwell times can be considered 'macro-fixations', in that they organize fixations into a larger picture.

For temporal characteristics, we include two criteria: whether the algorithm uses duration information, and whether the algorithm is locally adaptive. The use of duration information is guided by the fact that fixations are rarely less than 100 ms and often in the range of 200-400 ms. The incorporation of local adaptivity allows the interpretation of a given data point to be influenced by the interpretation of temporally adjacent points; this is useful, for instance, to compensate for differences between 'steady-eyed' individuals and those who show large and frequent eye movements.

3. REPRESENTATIVE ALGORITHMS

To enable us to rigorously compare the classes of algorithms in our taxonomy, it is convenient to formalize sample algorithms that serve to represent the essential ideas embodied by each class of algorithms. This section describes five algorithms— I-VT, I-HMM, I-DT, I-MST, and I-AOI—that extract the important aspects of numerous existing algorithms and express their basic techniques as simply as possible. Table 1 shows how the representative algorithms map onto the proposed taxonomy. Of course, there are many other possible algorithms that we could include in this analysis, such as hybrid algorithms with multiple heuristics; however, due to space constraints, we limit this exposition to five algorithms with fairly different approaches and implementations.

3.1 Velocity-Based Algorithms

3.1.1 Velocity-Threshold Identification (I-VT)

Velocity-threshold fixation identification (I-VT) is the simplest of the identification methods to understand and implement. I-VT is a velocity-based method that separates fixation and saccade points based on their point-to-point velocities [e.g., 5, 18]. The velocity profiles of saccadic eye movements show essentially two distributions of velocities: low velocities for fixations (i.e., <100 deg/sec), and high velocities (i.e., >300 deg/sec) for saccades. This aspect of saccadic eye movements makes velocity-based discrimination fairly straightforward and robust. While it is possible to create a locally adaptive velocity-based fixation identification algorithm, velocity profiles have strong physical and physiological underpinnings, and thus static criteria are usually sufficient here.

I-VT begins by calculating point-to-point velocities for each point in the protocol. Each velocity is computed as the distance between the current point and the next (or previous) point. I-VT then classifies each point as a fixation or saccade point based on a simple velocity threshold: if the point's velocity is below threshold, it becomes a fixation point, otherwise it becomes a saccade point. The process then collapses consecutive fixation points into fixation groups and discards saccade points. Finally, I-VT translates each fixation group to a representation $<x,y,t,d>$ using the centroid (i.e., center of mass) of the points as x and y, the time of the first point as t, and the duration of the points as d. Pseudocode for the I-VT method is shown in Table 2.

Table 2: Pseudocode for the I-VT algorithm.

```
I-VT (protocol, velocity threshold)

Calculate point-to-point velocities for each
point in the protocol

Label each point below velocity threshold as
a fixation point, otherwise as a saccade
point

Collapse consecutive fixation points into
fixation groups, removing saccade points

Map each fixation group to a fixation at the
centroid of its points

Return fixations
```

I-VT requires the specification of one parameter, the velocity threshold. If angular velocities can be computed (i.e., the distance from eye to visual stimuli is known), the point-to-point velocity threshold can be approximated from a reasonable angular velocity threshold [6]; for instance, Sen and Megaw [18] used a threshold of 20 degrees/second. However, when only point-to-point velocities are known, an appropriate value for the velocity threshold may need to be inferred based on aspects of data collection (e.g., sampling frequency) along with some exploratory data analysis.

3.1.2 HMM Identification (I-HMM)

Hidden Markov model fixation identification (I-HMM) uses probabilistic analysis to determine the most likely identifications for a given protocol [15, 16]. Hidden Markov models (HMMs) are probabilistic finite state machines that have been employed extensively in the fields of speech and handwriting recognition [see 13]. I-HMM uses a two-state HMM in which the states represent the velocity distributions for saccade and fixation points. This probabilistic representation helps I-HMM perform more robust identification than a fixed-threshold method such as I-VT.

The crux of I-HMM is the two-state HMM shown in Figure 1. The HMM includes two sets of probabilities: observation and transition probabilities. The observation probabilities for each state (the small v distributions) represent the distribution of expected velocities in that state. The first state represents saccade points, and thus contains a distribution centered around higher velocities; the second state represents fixation points, and thus contains a distribution centered around lower velocities. The transition probabilities for each state (the arrows exiting the states) represent the likelihood of remaining in the state or making a transition to another state. The transition probabilities in Figure 1 show a large likelihood of remaining in each state (.95) and a small likelihood or making a transition (.05). Thus, the HMM provides a probabilistic representation of the observations (i.e., velocities) generated during saccadic eye movements.
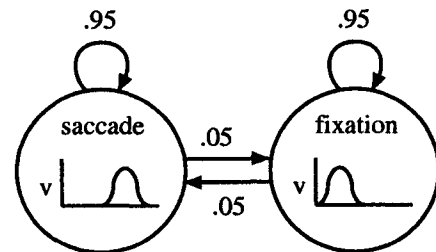

Figure 1: Sample two-state HMM. The first state represents higher-velocity saccade points; the second state represents lower-velocity fixation points.

Given the two-state HMM, I-HMM determines the most likely identification of each protocol point through a process of decoding. HMM decoding finds an assignment of protocol points to states that maximizes the probability of the protocol given the HMM, using dynamic programming [13] to find this optimal assignment efficiently. The assignment associates each point with a state, thus providing an identification of each point as a fixation or saccade point. Like I-VT, I-HMM then collapses consecutive fixation points into groups and outputs fixations at the centroid of the groups. Table 3 presents pseudocode for this I-HMM algorithm.

The parameters of I-HMM comprise the observation and transition parameters in the two-state HMM: two observation probability parameters (the mean and variance of the distribution) and two transition probabilities for each state, for a total of eight parameters. While this parameter space is

73

Table 3: Pseudocode for the I-HMM algorithm.

```
I-HMM (protocol, HMM)

Calculate point-to-point velocities for each
point in the protocol

Decode velocities with two-state HMM to
identify points as fixation or saccade points

Collapse consecutive fixation points into
fixation groups, removing saccade points

Map each fixation group to a fixation at the
centroid of its points

Return fixations
```

Table 4: Pseudocode for the I-DT algorithm.

```
I-DT (protocol, dispersion threshold,
duration threshold)

While there are still points

    Initialize window over first points to
    cover the duration threshold

    If dispersion of window points <=
    threshold

        Add additional points to the window
        until dispersion > threshold

        Note a fixation at the centroid of the
        window points

        Remove window points from points

    Else

        Remove first point from points

Return fixations
```

certainly more complex than that of I-VT, the parameters of I-HMM can be learned through a procedure called reestimation [13]. Given a training set (i.e., a set of protocols) reestimation learns the values of HMM parameters that maximize the probability of the training set given the HMM. Thus, with protocol data from a particular eye-tracking setup, reestimation can provide parameter values that, once estimated, can be used for any protocols collected in that setup.

3.2 Dispersion-Based Algorithms

3.2.1 Dispersion-Threshold Identification (I-DT)

In contrast to the velocity-based identification of I-VT and I-HMM, dispersion-threshold identification (I-DT) utilizes the fact that fixation points, because of their low velocity, tend to cluster closely together. I-DT identifies fixations as groups of consecutive points within a particular dispersion, or maximum separation [e.g., 19, 21]. Because fixations typically have a duration of at least 100 ms, dispersion-based identification techniques often incorporate a minimum duration threshold of

100-200 ms [21] to help alleviate equipment variability. The following I-DT algorithm is based on Widdel's [21] data reduction algorithm.

The I-DT algorithm uses a moving window that spans consecutive data points checking for potential fixations. The moving window begins at the start of the protocol and initially spans a minimum number of points, determined by the given duration threshold and sampling frequency. I-DT then checks the dispersion of the points in the window by summing the differences between the points' maximum and minimum x and y values; in other words, dispersion $D = [\max(x) - \min(x)] + [\max(y) - \min(y)]$. Note that alternative dispersion metrics could be based upon spatial variance or area of samples. If the dispersion is above the dispersion threshold, the window does not represent a fixation, and the window moves one point to the right. If the dispersion is below the dispersion threshold, the window represents a fixation. In this case, the window is expanded (to the right) until the window's dispersion is above threshold. The final window is registered as a fixation at the centroid of the window points with the given onset time and duration. This process continues with window moving to the right until the end of the protocol is reached. Table 4 includes pseudocode for the I-DT algorithm.

We can note that this characterization of fixations uses the centroid and diameter. A circular area is usually assumed, and the mean distance from each sample to the fixation centroid provides an estimate of the radius. Also, dispersion-based algorithms are sometimes used to locate clusters within minimum spanning tree network representations (see the following section). The graph forms an efficient framework for rapid search of large sample sets.

The I-DT algorithm requires two parameters, the dispersion threshold and the duration threshold. Like the velocity threshold for I-VT, the dispersion threshold can be set to include 1/2° to 1° of visual angle if the distance from eye to screen is known. Otherwise, the dispersion threshold can be estimated from exploratory analysis of the data. The duration threshold is typically set to a value between 100 and 200 ms [21], depending on task processing demands.

Note that this characterization of I-DT is conceptually different from some dispersion-based methods that center on clustering algorithms. For instance, the leader algorithm and the k-means algorithm are two such methods [8]. However, most of these algorithms require an initial estimate of the number of clusters in the protocol, making them less useful for eye-movement data, where the number of fixations is typically unknown.

3.2.2 MST Identification (I-MST)

MST identification (I-MST) is based on minimum spanning trees (MSTs) — that is, a tree connecting a set of points such that the total length of the tree's line segments is minimized. MSTs can provide a highly flexible and controllable representation for dispersion-based fixation identification [7]. A two-step approach is necessary, first requiring the construction of the MST followed by a search of the MST. Construction uses Prim's algorithm [2]. There is one and only one MST for a set of points. The advantage of an MST data

Table 5: Pseudocode for the I-MST algorithm.

```
I-MST (protocol, edge ratio, edge sd)

Construct MST from protocol data points using
Prim's algorithm

Find the maximum branching depth for each MST
point using a depth-first search

Identify saccades as edges whose distances
exceed predefined criteria

Define the parametric properties (μ, σ) of
local edges, identifying saccades when an
edge length exceeds a defined ratio

Identify fixations as clusters of points not
separated by saccades

Return fixations
```

Table 6: Pseudocode for the I-AOI algorithm.

```
I-AOI (protocol, duration threshold,
target areas)

Label each point as a fixation point for the
target area in which it lies, or as a saccade
point if none

Collapse consecutive fixation points for the
same target into fixation groups, removing
saccade points

Remove fixation groups that do not span the
minimum duration threshold

Map each fixation group to a fixation at the
centroid of its points

Return fixations
```

representation lies in the degree of control, flexibility, and local adaptation for dispersion analysis, as well as improved subsequent characterizations of defined fixations.

Fixation identification requires traversing the already defined MST, as described in Table 5. A depth-first search is made to determine the maximum depth of interconnectivity at each point. Branching depths below a defined setpoint signal locations near the edge of the MST that are not appropriate candidates for separating fixations. If the edges connected to each endpoint exceed the defined minimum branching depth, then the network of edges connected to each endpoint is gathered into an associated edge length distribution. The mean μ and standard deviation σ of edge lengths provide a locally adaptive comparison for separation of fixations. Separation can occur based upon comparison of the edge under consideration of both μ and σ of neighbor edge lengths. In this way, a framework is achieved for controlling where fixations may occur in an MST and for determining how local adaptivity affects fixation decisions.

The MST format allows additional characterization parameters for fixations. For example, the MST length is defined by the longest path through the graph. Critical paths can be defined by areas with minimal branching structure. These and other parameters allow an estimate of the shape and possibly direction of fixations; these go beyond simpler centroid and size characterizations.

3.3 Area-based Algorithms

3.3.1 Area-of-Interest Identification (I-AOI)

The four previous identification methods can identify fixations at any location in the visual field. In contrast, area-of-interest fixation identification (I-AOI) identifies only fixations that occur within specified target areas [e.g., 3]. The target areas are rectangular regions of interest that represent units of information in the visual field. These target areas, generally used in later analyses like tracing, keep identified fixations close to relevant targets. I-AOI also utilizes a duration threshold to help distinguish fixations in target areas from passing saccades in those areas.

I-AOI begins by associating data points with target areas: it labels points within a target area as a fixation point for that target, and labels points outside of all target areas as saccades. I-AOI then collapses consecutive fixation points for the same target into fixation groups, discarding saccade points. Finally, it removes fixation groups that fall below a given duration threshold and transforms each fixation group into a fixation tuple. The pseudocode for this algorithm appears in Table 6.

As explained earlier, I-AOI can provide a larger picture than fixations alone. Like a fixation, dwell time within an area has a starting time and an ending time. However, since fixations can serve as the input data into AOI determination, the time between these dwells do not describe saccadic behavior. Rather, there could be multiple saccades and fixations interspersed between AOIs. While AOI-based dwell-time algorithms are not fixation algorithms per se, they are valuable as a concept to help explain higher-level collections of fixations organized about visual targets and areas.

4. EVALUATION AND COMPARISON

To evaluate and compare the various identification methods, we now consider each method with respect to several characteristics: interpretation speed, accuracy, robustness, ease of implementation, and parameter setting. Table 7 shows a summary of the described methods according to these characteristics. To illustrate some of the issues discussed, we utilize a sample eye-movement protocol from a task in which students encoded four equation values and performed some simple computation [see 15, 16], as shown in Figure 2. Each point represents the student's current point-of-regard sampled at a rate of 60 Hz. Points that represent fixations, as interpreted by the various identification algorithms, are drawn larger than those that represent saccades. Also, points that occur earlier in the protocol time sequence are drawn darker than those that occur later. The fixations are numbered to facilitate understanding of the protocol time sequence.

Method	Accuracy	Speed	Robustness	Impl. Ease	Parameters
Velocity Threshold (I-VT)	√	√√	×	√√	1
Hidden Markov Model (I-HMM)	√√	√	√√	√/×[a]	8 / 0[a]
Dispersion Threshold (I-DT)	√√	√	√√	√	2
Minimum Spanning Tree (I-MST)	√	×	√√	×	2
Area-of-Interest (I-AOI)	×	√	√	√	1+[b]

Key: √√ = very good, √ = good, × = not as good.
[a] I-HMM has 8 parameters in its two-state HMM that can be an be learned through reestimation. Without reestimation, I-HMM has 8 parameters but is simpler to implement; with reestimation, I-HMM effectively has no parameters but is more difficult to implement.
[b] I-AOI has 1 parameter but also requires specification of target areas.

4.1 Velocity-Based Methods

I-VT is straightforward to implement, runs very efficiently, and can easily run in real time. However, it sometimes encounters problems when point velocities hover near threshold because of eye-tracker noise or time-averaged data. Specifically, when point velocities are near threshold (e.g., midway between saccades and fixations), the strict threshold can result in "blips" in identification—fixation or saccade groups with only one or a few consecutive points. The protocol in Figure 2(a) illustrates this problem: fixations 3-6 are interpreted as four separate fixations, whereas it seems more likely that there are only one or two actual fixations represented in these data points. Similarly, fixations 11-13 seem to represent a single fixation. The problem is especially prevalent when analyzing time-averaged data, such as those in the figure. Researchers have alleviated the problem in the past by aggregating consecutive fixations over a single target into *gazes* on that target [9], as shown in Figure 2(b). Other work required minimum durations for either fixations or saccades; for instance, Sen and Megaw [18] required that saccade points remain above a velocity threshold for at least 10 ms.

I-HMM employs a probabilistic model rather than a fixed velocity threshold and utilizes the sequential information in the protocol. Thus, I-HMM allows for more freedom in identifying points and provides more robust analysis than I-VT, especially in the presence of significant noise. Figure 2(c) illustrates the same protocol as in Figure 1 analyzed using I-HMM; the problematic fixations that I-VT separated into multiple fixations (see Figure 2(a)) are grouped into more plausible fixations in Figure 2(c). Another benefit of I-HMM is that the two-state HMM can be embedded into a larger cognitive model for even more robust interpretation of eye movements [15, 16]. In addition, I-HMM, like I-VT, runs efficiently in linear time [see 15, 16] and can also run in real time. The primary disadvantage of I-HMM is the difficulty of implementing the reestimation procedure, which is both complex and tedious. However, if no reestimation of parameters is needed, the I-HMM algorithm alone (including HMM decoding) is fairly straightforward to implement.

4.2 Dispersion-Based Methods

I-DT is a linear-time, potentially real-time algorithm that produces robust identification results. For the sample protocol in Figure 2(a), it generates an almost identical analysis as I-HMM (as shown in Figure 2(c)), avoiding the "blip" problem because of its duration threshold. The primary disadvantage of I-DT is the use of two parameters that are highly interdependent; for instance, a small dispersion threshold with a large duration threshold may not result in any identified fixations. Thus, I-DT requires the most careful parameter setting but provides good results for a fairly simple algorithm.

I-MST has a somewhat different flavor than the other algorithms in the sense that it forms clusters of fixation points without using sequential information in the protocols. Thus, it is more difficult to compare with the other algorithms. I-MST allows for control of parameter settings that define a saccade by varying the relative importance of the edge length ratio and mean/variance tests for clustering. Its speed is exponential in that it becomes exponentially slower as points are added: it requires multiple traversals over the MST for each point, one to establish local edge distribution characteristics and one to test each potential cut edge. It is also more difficult to make I-MST a real-time algorithm because of its global analysis of existing data points. However, I-MST also seems to provide additional means of characterizing fixations beyond spatial location and size; for instance, the MST diameter provides the longest MST path within each fixation. Also, characteristics of the MST, such as the ratio of path length to width, can provide information on the shape of the defined fixations. Branching depth provides an additional means of assessing fixation size.

4.3 Area-Based Methods

I-AOI runs efficiently in linear time or real time. However, it suffers from a somewhat serious problem: the inclusion of many saccade points in identified fixations. Because any point within a target area classifies as a fixation point, saccade points leading into a fixation are often included in the final fixation. This problem can result in deceptively long durations for identified fixations. An even more problematic situation
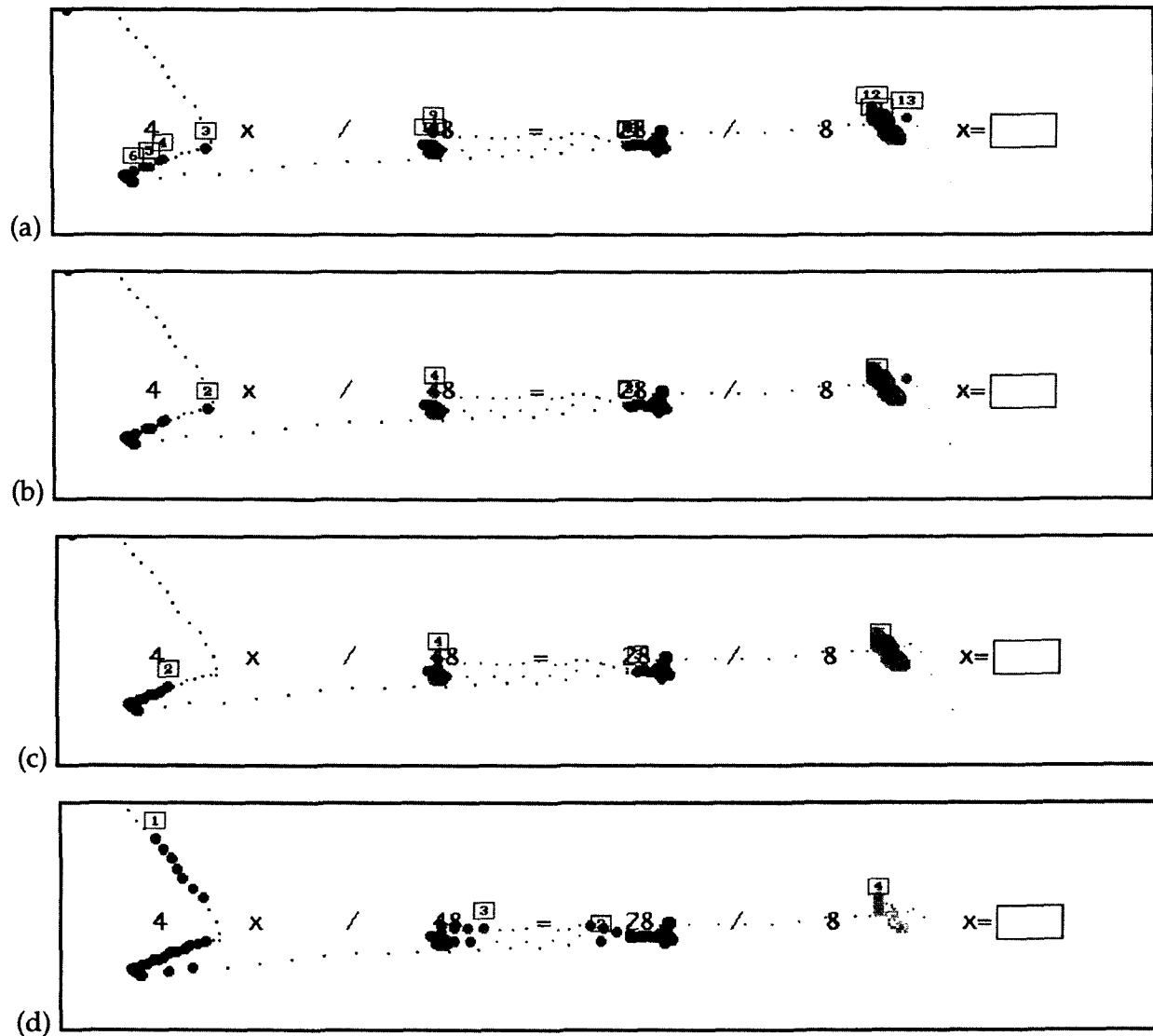
Figure 2: Fixation identification using (a) I-VT, (b) I-VT with fixations collapsed into gazes, (c) I-HMM or I-DT, and (d) I-AOI. Eye-movement protocols are taken from an equation-solving task in which a student encodes the four equation values and computes a result [see 15, 16]. Larger points represent fixations while smaller points represent saccades (as interpreted by the identification algorithm). Darker points occur earlier while lighter points occur later in the protocol.

arises when long saccades through large target areas are identified as fixations, especially when the data include some form of time averaging. Figure 2(d) illustrates these problems: fixation 1 includes points that likely represent a saccade (the upper grouping of points), whereas fixation 4 fails to identify apparent fixation points because they lie outside the target areas used. Overall, I-AOI can provide reasonable results [e.g., 3] for certain aggregate analyses but is generally less preferable than the other, more robust algorithms.

5. SUMMARY AND CONCLUSIONS

To summarize our qualitative analysis, I-HMM and I-DT provide accurate and robust fixation identification by incorporating sequential information to aid in interpretation. I-MST also provides robust identification but runs slower than all other algorithms. I-VT has the simplest algorithm as thus the smallest computational overhead; however, it can experience severe "blip" effects when analyzing at the level of fixations rather than gazes. I-AOI performs rather poorly on all fronts for the purpose of identification and is best not used.

These results offer several implications for future use of these and related algorithms. First, velocity-based and dispersion-based algorithms both fare well and provide approximately equivalent performance. However, area-based algorithms are too restrictive and can generate deceptive results that bias later analyses. Second, the use of temporal information can greatly facilitate fixation identification of protocols. The three

algorithms that utilize this information in a locally adaptive way, namely I-HMM, I-DT, and I-MST, generate robust interpretations even in the presence of eye-tracking equipment noise. Third, in support of work such as Karsh and Breitenbach [10], the choice of identification algorithms can dramatically affect the resulting identified fixations. By describing their chosen algorithm in terms of the proposed taxonomy, researchers can better communicate their analysis techniques to others and thus provide more standardized and understandable results.

As the next step in evaluating and comparing fixation identification algorithms, we hope to generate more quantitative measures for comparison and examine how the various algorithms affect later analyses in the context of different domain applications. For instance, some preliminary work has shown that a subset of the algorithms described here can affect the interpretation of eye-tracking protocols with respect to the predictions of a cognitive process model [15, 16]. This increased understanding of possible identification algorithms will help to facilitate the use and analysis of eye-tracking data for both scientific research and the development of eye-based systems and applications.

REFERENCES

[1] Alpern, M. (1962). Type of movement. In H. Davson (Ed.), *The Eye*, Vol. 3. (pp. 63-151). New York: Academic Press.

[2] Camerini, P. M., Galbiati, G., & Maffioli, F. (1988). Algorithms for finding optimum trees: Description, use and evaluation. *Annals of Operations Research, 13*, 265-397.

[3] DenBuurman, R., Boersma, T., & Gerrisen, J. F. (1981). Eye movements and the perceptual span in reading. *Reading Research Quarterly, 16*, 227-235.

[4] Ditchburn, R. W. (1980). The function of small saccades. *Vision Research, 20*, 271-272.

[5] Erkelens, C. J., & Vogels, I. M. L. C. (1995). The initial direction and landing position of saccades. In J. M. Findlay, R. Walker, & R. W. Kentridge (Eds.), *Eye Movement Research: Mechanisms, Processes, and Applications* (pp. 133-144). New York: Elsevier.

[6] Fuchs, A. F. (1971). The saccadic system. In P. Bach-y-Rita, C. C. Collins, & J. E. Hyde (Eds.), *The Control of Eye Movements* (pp. 343-362). New York: Academic Press.

[7] Goldberg, J. H., & Schryver, J. C. (1995). Eye-gaze-contingent control of the computer interface: Methodology and example for zoom detection. *Behavior Research Methods, Instruments, & Computers, 27*, 338-350.

[8] Hartigan, J. A. (1975). *Clustering algorithms.* New York: John Wiley & Sons.

[9] Just, M. A., & Carpenter, P. A. (1984). Using eye fixations to study reading comprehension. In D. E. Kieras & M. A. Just (Eds.), *New Methods in Reading Comprehension Research* (pp. 151-182). Hillsdale, NJ: Erlbaum.

[10] Karsh, R., & Breitenbach, F. W. (1983). Looking at looking: The amorphous fixation measure. In R. Groner, C. Menz, D. F. Fisher, & R. A. Monty (Eds.), *Eye Movements and Psychological Functions: International Views* (pp. 53-64). Hillsdale, NJ: Erlbaum.

[11] Land, M. F., & Lee, D. N. (1994). Where we look when we steer. *Nature, 369,* 742-744.

[12] Noton, D., & Stark, L. (1971). Scanpaths in saccadic eye movements while viewing and recognizing patterns. *Vision Research, 11,* 929-942.

[13] Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE, 77,* 257-286.

[14] Rayner, K. (1995). Eye movements and cognitive processes in reading, visual search, and scene perception. In J. M. Findlay, R. Walker, & R. W. Kentridge (Eds.), *Eye Movement Research: Mechanisms, Processes, and Applications* (pp. 3-21). New York: Elsevier.

[15] Salvucci, D. D. (1999). *Mapping eye movements to cognitive processes.* Doctoral Dissertation, Department of Computer Science, Carnegie Mellon University.

[16] Salvucci, D. D., & Anderson, J. R. (1998). Tracing eye movement protocols with cognitive process models. In *Proceedings of the Twentieth Annual Conference of the Cognitive Science Society* (pp. 923-928). Hillsdale, NJ: Erlbaum.

[17] Salvucci, D. D., & Anderson, J. R. (in press). Integrating analogical mapping and general problem solving: The path-mapping theory. *Cognitive Science.*

[18] Sen, T., & Megaw, T. (1984). The effects of task variables and prolonged performance on saccadic eye movement parameters. In A. G. Gale & F. Johnson (Eds.), *Theoretical and Applied Aspects of Eye Movement Research* (pp. 103-111). Amsterdam: Elsevier.

[19] Stark, L., & Ellis, S. R. (1981). Scanpath revisited: Cognitive models of direct active looking. In D. F. Fisher, R. A. Monty, & J. W. Senders (Eds.), *Eye Movements: Cognition and Visual Perception* (pp. 193-226). Hillsdale, NJ: Erlbaum.

[20] Suppes, P. (1990). Eye-movement models for arithmetic and reading performance. In E. Kowler (Ed.), *Eye Movements and their Role in Visual and Cognitive Processes* (pp. 455-477). New York: Elsevier.

[21] Widdel, H. (1984). Operational problems in analysing eye movements. In A. G. Gale & F. Johnson (Eds.), *Theoretical and Applied Aspects of Eye Movement Research* (pp. 21-29). New York: Elsevier.