Ferns for area of interest free scanpath classification

Wolfgang Fuhl  
University Tübingen, Perception Engineering  
Germany  
wolfgang.fuhl@uni-tuebingen.de

Nora Castner  
University Tübingen, Perception Engineering  
Germany  
castner@informatik.uni-tuebingen.de

Thomas Kübler  
University Tübingen, Perception Engineering  
Germany  
thomas.kuebler@uni-tuebingen.de

Alexander Lotz  
Daimler AG, Truck Product Engineering  
Stuttgart  
rene_alexander.lotz@daimler.com

Wolfgang Rosenstiel  
University Tübingen, Technical Computer Science  
Germany  
Wolfgang.Rosenstiel@uni-tuebingen.de

Enkelejda Kasneci  
University Tübingen, Perception Engineering  
Germany  
enkelejda.kasneci@uni-tuebingen.de

ABSTRACT
Scanpath classification can offer insight into the visual strategies of groups such as experts and novices. We propose to use random ferns in combination with saccade angle successions to compare scanpaths. One advantage of our method is that it does not require areas of interest to be computed or annotated. The conditional distribution in random ferns additionally allows for learning angle successions, which do not have to be entirely present in a scanpath. We evaluated our approach on two publicly available datasets and improved the classification accuracy by \( \approx 10 \) and \( \approx 20 \) percent.

CCS CONCEPTS
• Computer systems organization → Embedded systems; Redundancy; Robotics; • Networks → Network reliability;

KEYWORDS
Eye tracking, scanpath analysis, random ferns

ACM Reference Format:

1 INTRODUCTION
The way we direct our eyes at can tell us much more than what we are looking at. Naturally, gaze behavior reflects the interplay of cognitive as well as sensory processes. Consequently, patterns of fixations and saccades, known as the scanpath, can offer insight into the order and nature of information processing. Over multiple domains, eye tracking studies have identified patterns in gaze behavior. The literature found that these patterns could characterize specific subject groups (e.g., novices and experts), or experimental settings, such as the task assigned to a subject.

For instance in art, eye scanning differences have been found between professional and naïve art viewers for both realistic and abstract art [Zangemeister et al. 1995]. Also, top-down beliefs as well as bottom-up feature attention can affect the gaze behavior on artworks [Locher et al. 2015; Massaro et al. 2012].

Additionally in the medical domain, scanpath differences can reflect the professional as well as the treatment aspects. Scanpath differences between novices and experts have been found in microsurgical surgeons [Eivazi et al. 2012; Kübler et al. 2015a] as well as radiologists [Manning et al. 2006; Van der Gijp et al. 2017]. It was also found that dental students who received a specialized radiography training course could be accurately identified from their scanpaths [Castner et al. 2018]. On the treatment aspect, gaze behavior differences from healthy controls have been measured in both patients suffering from schizophrenia [Loughland et al. 2002] and autism spectrum disorder [Horley et al. 2004; Pelphrey et al. 2002]. Hence, scanpaths can likely be employed towards more refined training, diagnostic, and treatment protocols.

In driving, scanpaths have been used to robustly determine safe or unsafe driving in people with visual field defects [Kasneci et al. 2014; Kübler et al. 2015b]. In addition, they can be used in driver assistance systems to indicate the take-over readiness [Braunagel et al. 2017] or cognitive load [Palinko et al. 2010], and fatigue [Ji et al. 2004].

Interestingly, most of the studies mentioned above concentrate on finding statistically significant differences in individual scanpath metrics. Therefore, there is a large and ever growing body of scanpath comparison and classification methodology: Ranging from simple statistics to state-of-the-art machine learning (see [Anderson et al. 2015] for a review).

2 RELATED WORKS
In 1935, Buswell hypothesized that “the mental set obtained by the directions given [...] obviously influences the characteristics of the perceptual process” [1935]. A finding followed up in late
sixties/early seventies by Yarbus [1967] and Noton & Stark [1971] when they measured gaze pattern differences as an effect of contextual information. Here, the term scanpath was initially used, and the method for comparing multiple scanpaths was executed manually: Using semantic understanding of the image. Even today, some aspects of manual scanpath comparison are still used; such as AOs hand-labeled by the experimenter, which can be tedious, task dependent, and subjective [Jarodzka et al. 2010].

Only in the nineties were automated metrics initially proposed [Brandt and Stark 1997]. Since then, a wide array of methodology for automated scanpath comparison has evolved [Anderson et al. 2015]. More recently, machine-learning based approaches with impressive results have emerged [Crabb et al. 2014; French et al. 2017; Hoppe et al. 2018; Kübler et al. 2017; Zhang and Le Meur 2018]. In general, they can handle the task of distinguishing relevant eye movement patterns from an overall high level of noise. However, how the eye movement trajectories should be encoded to enable efficient machine learning is still an area of open debate. While some algorithms rely heavily on a massive agglomeration of time-aggregated features or complete-sequence alignment [Burch et al. 2018; Cristino et al. 2010; Dewhurst et al. 2018; Hoppe et al. 2018], the use of gaze transitions (i.e., the shift of gaze between two targets) has become one of the most popular features. This strategy is notable, as a chain of cognitive associations between gaze targets can be modeled in this way. Also, while Hidden Markov Models (HMMs) are still the most common approach [Coutrot et al. 2018; Ellis and Stark 1986; Hacisalihzade et al. 1992], other methods have emerged that extend these patterns in length to span multiple subsequent fixations and saccades. Finding patterns in such lengthier sequences is of special importance, as the specificity of these patterns for a specific task or subject group is likely increased and, therefore, they are highly useful for the classification task [Kübler et al. 2017].

3 METHOD
We analyzed the patterns that represent the angles between successive saccades. An example is shown in Figure 1, where similar angle patterns are a subsequence, as apparent over multiple participants. Thus, our algorithm searches for repetitive patterns over subjects. These repetitions are then used to classify the scanpath into a category. Since such patterns are subject to large variations between individuals and even between repeated trials—not to mention eye tracker inaccuracies—comparing angular patterns requires a slack range in which two angles are considered similar. Therefore, each angle in a pattern has an assigned and variable angular tolerance range. By employing ferns [Bosch et al. 2007; Ozuysal et al. 2010], we can allow assigning probability values to each permutation of such a pattern, which makes it possible to correctly classify incomplete patterns. In the following section, each step of our approach is described in detail.

3.1 Saccade sequence angles

Figure 2 shows two possible approaches to compute the angles between saccades from a sequence of saccades. The major difference between these approaches is whether the angles are calculated relative to an absolute reference frame or relative to the preceding saccade. Both can likely be valid approaches. The latter adds invariance to rotations of the pattern, which could be attractive for data from head-mounted eye-trackers. Whereas the former is able to distinguish between rotated representations of the pattern.

In our evaluation, both approaches performed almost equally for both data sets evaluated. Since the angle between saccades is invariant to rotation of the eye tracker—a challenge especially with head-mounted eye trackers—we believe it is the more robust approach. However, our implementation provides both computation models. In the following, these angles will be denoted as $\alpha_i f_i$, with the indices $i$ and $f_i$ denoting the index and the assigned fern index (i.e., the position index in the fern where it is used), respectively.

Figure 3 shows the angular tolerance ranges in which saccades would be considered similar to the current pattern. In our implementation, those tolerances can be in the range of $0^\circ$ to $359^\circ$. Thus, a
The first index \( i \) specifies the sequence of conditions that together define the probability for class \( k \). This probability relates directly to the probability that the seen features are evaluated by the fern and matched to the input vector \( F \).

These features are evaluated by the fern and matched to the input vector \( F \). For a saccade sequence to match a specific pattern/feature vector, a conditional distribution (indicated by the black bars in figure 5). Where the features on the scanpaths. In the case that a feature is successfully matched, it is encoded by a 1, otherwise by a 0 at the respective fern index. This step builds a binary-encoded number, which is used as an index to the probability distribution.

A set of features is evaluated by the fern and matched to the input vector \( F \). For a saccade sequence to match a specific pattern/feature vector, a conditional distribution (indicated by the black bars in figure 5). Where the features on the scanpaths. In the case that a feature is successfully matched, it is encoded by a 1, otherwise by a 0 at the respective fern index. This step builds a binary-encoded number, which is used as an index to the probability distribution.

A uniform distribution must be assumed for initialization, since several of the ferns in combination serve as classifiers and the probabilities of the individual ferns are multiplied. If this step is not performed, zero probabilities can occur, which can influence the classification negatively.

### 3.3 Training

The training of ferns consists of three parts, namely feature selection, distribution training, and fern combination selection. Due to the massive amount of possible features, brute-force for all possible patterns is only applicable for short feature length. Therefore, we only compute features up to a length of three and only inspect angle successions that occur in the training data scanpaths.

First, we extract all possible angles of length one to three. Since a feature consists of tuples \(((\alpha, \rho)_i, f_i)\), we have to assign each angle \((\alpha_i, f_i)\) a range \((\rho_i, f_i)\). This assignment is done by inspecting all possible ranges per angle and evaluating those on the training dataset with a step width of one degree: Running in polynomial time \((360^{\text{length}})\). We define this evaluation as the feature score.

\[
FS_j = | \sum (\alpha, \rho)(T_j) = 0 - \sum (\alpha, \rho)(T_j) = 1 | \tag{1}
\]
For each feature and each class \( (j) \), this score can be computed using Equation 1. Features for one specific class that result in a score that deviates by 10% from the average score of all classes were added to the feature set of that class. Afterward, the features are sorted based on this score to select the best features directly.

The second part of our training is the selection of a set of ferns. The conditional distribution size per fern is equal to the number of features, and the amount of distributions is equal to the class labels in the training set. Therefore, if seven classes have to be distinguished, each distribution has a size of \( 2^7 \), with \( F \) denoting the amount of used features, and each fern uses seven of these distributions. The 2 is due to each feature evaluating as either one or zero. Afterward, we select one feature per class from the feature pool and add it to the feature set of the fern. The conditional distribution is filled with a uniform distribution and trained on the training dataset based on the occurrence of the different combinations statistically. Since this step does not necessarily result in an optimal feature combination, we evaluate all possible and select the best. This process can also be done in polynomial time \((\text{featurepoolsize})^{\text{classlabels}}\), since the class labels are fixed. In our implementation, we compute this step using threads to reduce the training time.

The third and last step of the training is the selection of a combination of multiple ferns. The final classification is performed by multiplication of all fern probabilities and selection of the highest ranking probability as the final label. Since we have already evaluated all possible combinations of features per fern, which results in a fern pool to directly select from. Here, we inspect the probability per class, which our existing detector provides in comparison to the possible new fern in the detector. For the calculation, the existing detector consisting of several ferns is evaluated on the training data as well as the possible new fern. Afterward, the average difference between both results is computed and the fern with the highest difference is selected as the new member of the detector. This process starts with the selection of the best fern and ends after ten ferns are selected. The amount of ten ferns was selected empirically and can be changed in our implementation.

### 4 EVALUATION

For the evaluation and comparison to the state-of-the-art, we used two publicly available data sets. The first data set is the task classification in the second experiment from the Defending Yarbus (DY) paper [Borji and Itti 2014]. It contains seven tasks and 45 recordings for each of them: Data is provided as fixation coordinates. The second data set is the task classification from Reconsidering Yarbus (RY) paper [Greene et al. 2012], in which four tasks were distinguished and \( \approx 80 \) recordings are provided per task: Data is provided as gaze points. Since our algorithm needs fixations as input, we computed the mean velocity for the gaze points per trial and used this value as a threshold to separate fixations and saccades. This method is not an accurate event detection, but was performed to make the results easily reproducible without relying on algorithm and parameter choices for event detection. As our focus is identifying each saccade and not about e.g. accurate fixation duration, such a coarse–but simple–event detection is sufficient for this application. For each data set, we performed ten fold cross-validation; each fold contained an equal amount of recordings for each task. Therefore, each fold in DY dataset contained four recordings of each task and seven recordings for RY dataset. The non-integer divisible share was added to the training data. Table 1 shows our results in comparison to the state-of-the-art. As evident, our approach outperforms the competitor algorithms. Moreover, our algorithm does not require areas of interest (AOIs). This limitation to scanpath algorithms was already treated in [Kübler et al. 2017], in which the AOIs are computed based on the data. Since both angle calculation functions deliver approximately the same results, it is not possible to make a statement regarding the robustness based on this evaluation (Table 1); further research has to be conducted to evaluate this aspect. One disadvantage of our algorithm is the long training time due to the polynomial complexity of selecting suitable patterns. However, it can be solved by selecting only a subset of the training data. Another issue is that long recordings will require a windowing function.

### 5 CONCLUSION

We proposed a novel algorithm for scanpath classification. It uses angle and angle-range tuples as features and assigns probabilities based on conditional distributions. Our algorithm outperformed the state-of-the-art on two publicly available datasets. However, the disadvantages of our algorithm are the polynomial complexity of the training and therefore, the high computational demand.

Since our approach is independent of AOIs and the features are based on angles that are calculated between saccades, the next planned step is to create a data set using head-mounted eye trackers. This data would further elaborate and verify that angle computation method between saccades and whether rotation invariance is of importance for data including head movements. In addition, we intend to evaluate optimal time segment sizes based on eye tracker frame rates as well as for different tasks, e.g., expert and novices, driver state detection, etc.

### ACKNOWLEDGMENTS

Work of the authors is supported by the Institutional Strategy of the University of Tübingen (Deutsche Forschungsgemeinschaft, ZUK 63)

---

**Table 1: The classification accuracy in % of our approach in comparison to the state-of-the-art for task classification.**

<table>
<thead>
<tr>
<th>Method</th>
<th>DY (7 Tasks)</th>
<th>RY (4 Tasks)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chance level</td>
<td>14.3</td>
<td>25.0</td>
</tr>
<tr>
<td>Fern(_{\text{ANG}-\text{X}})</td>
<td>43.57</td>
<td>54.06</td>
</tr>
<tr>
<td>Fern(_{\text{ANG}-\text{V}})</td>
<td>41.42</td>
<td>54.37</td>
</tr>
<tr>
<td>[Kübler et al. 2017]</td>
<td>24.2</td>
<td>34.4</td>
</tr>
<tr>
<td>[Borji and Itti 2014]</td>
<td>28.71</td>
<td>34.1</td>
</tr>
<tr>
<td>[Greene et al. 2012]</td>
<td>-</td>
<td>27.1</td>
</tr>
<tr>
<td>[Kanan et al. 2014]</td>
<td>-</td>
<td>33.0</td>
</tr>
</tbody>
</table>
REFERENCES


