Performance Metrics for Activity Recognition

JAMIE A. WARD
Lancaster University
PAUL LUKOWICZ
University of Passau
and
HANS W. GELLERSEN
Lancaster University

In this article, we introduce and evaluate a comprehensive set of performance metrics and visualisations for continuous activity recognition (AR). We demonstrate how standard evaluation methods, often borrowed from related pattern recognition problems, fail to capture common artefacts found in continuous AR—specifically event fragmentation, event merging and timing offsets. We support our assertion with an analysis on a set of recently published AR papers. Building on an earlier initial work on the topic, we develop a frame-based visualisation and corresponding set of class-skew invariant metrics for the one class versus all evaluation. These are complemented by a new complete set of event-based metrics that allow a quick graphical representation of system performance—showing events that are correct, inserted, deleted, fragmented, merged and those which are both fragmented and merged. We evaluate the utility of our approach through comparison with standard metrics on data from three different published experiments. This shows that where event- and frame-based precision and recall lead to an ambiguous interpretation of results in some cases, the proposed metrics provide a consistently unambiguous explanation.

Categories and Subject Descriptors: I.5.2 [**Pattern Recognition**]: Design Methodology

General Terms: Performance, Standardization

Additional Key Words and Phrases: Activity recognition, metrics, performance evaluation

ACM Reference Format:

Ward, J. A., Lukowicz, P., and Gellersen, H. W. 2011. Performance metrics for activity recognition. ACM Trans. Intell. Syst. Technol. 2, 1, Article 6 (January 2011), 23 pages.
DOI = 10.1145/1889681.1889687 http://doi.acm.org/10.1145/1889681.1889687

6

Authors' addresses: J. A. Ward and H. W. Gellersen, Lancaster University, Bailrigg, Lancaster. UK LA1 4YW; email: j.a.ward@lancaster.ac.uk, hwg@comp.lancs.ac.uk; P. Lukowitz, University of Passau, University of Passau, D-94030 Passau, Germany; email: paul.lukowicz@uni-passau.de.

DOI 10.1145/1889681.1889687 http://doi.acm.org/10.1145/1889681.1889687

1. INTRODUCTION

Human activity recognition (AR) is a fast-growing research topic with many promising real-world applications. As it matures so does the need for a comprehensive system of metrics that can be used to summarize and compare different AR systems. A valid methodology for performance evaluation should fulfil two basic criteria:

(1) It must be objective and unambiguous. The outcome of an evaluation should not depend on any arbitrary assumptions or parameters.
(2) It should not only grade, but also characterize performance. When comparing systems the method should give more then a binary decision, such as "A is better then B". Instead it should quantify the strengths and weaknesses of each and give the system designer hints as to how improvements can be made.

Ward et al. [2006a] demonstrated that the standard evaluation metrics currently used in AR do not adequately characterize performance. Information about typical characteristics of activity events are routinely ignored in favor of making recognition results fit standard metrics such as event or frame accuracy. For example, existing metrics do not reveal whether an activity has been fragmented into several smaller activities, whether several activities have been merged into a single large activity; or whether there are timing offsets in the recognition of an activity. This can lead to a presentation of results that can be confusing, and even misleading. As we will show in this article, this is not just a theoretical problem but an issue routinely encountered in real applications.

The problem of how to handle inexact time matching of ground truth to output has been identified in a range of AR research, with a typical solution being to ignore any result within a set margin of the event boundaries [Bao and Intille 2004], or to employ some minimum coverage rule [Tapia et al. 2004; Westeyn et al. 2005; Fogarty et al. 2006]. The problem of fragmented output has been noted in a handful of publications, with solutions ranging from treating fragments as correct events [Fogarty et al. 2006], to incorporating them in an equal way to insertion and deletion errors (e.g., "reverse splicing" [Patterson et al. 2005]). Evidence of merging was hinted at by Lester et al. [2005], and is discussed as an "episode spanning two activities," by Buettner et al. [2009].

In a first attempt at characterizing AR performance, Ward et al. [2006a] introduced an unambiguous method for calculating insertions and deletions alongside four new types of error: fragmentation, merge and the timing offset errors of overfill and underfill. Corresponding frame-by-frame metrics derived from all of these categories were also proposed alongside a convenient visualisation of the information. Although used in a handful of subsequent publications [Bulling et al. 2008; Minnen et al. 2007; Stiefmeier et al. 2006; Ward et al. 2006b], the original metrics suffer from a number of shortcomings:

(1) visualisation of frame errors using the error division diagram (EDD), which plots insertion, deletion, fragmenting, merge, correct and timing errors as a percentage of the total experiment time, is influenced by changes in the

proportion of different classes, or class skew. This makes comparability between datasets difficult.

(2) event errors were not represented in a metric format suitable for comparison. Instead absolute counts of insertions, deletions, etc., were shown.

This article extends the previous work in four ways, specifically we: (1) introduce a system of *frame-by-frame* metrics which are invariant to class skew and (2) introduce a new system of metrics for recording and visualising *event* performance. We then (3) apply the metrics to three previously published data sets, and (4) show how these offer an improvement over traditional metrics. The contributed methods are based on sequential, segment-wise comparison, but it is worth noting that they also have a significant amount of tolerance against small time shifts in the recognition. Unlike in other approaches (e.g., dynamic time warping, DTW [Berndt and Clifford 1994]), the time shift is not masked (or hidden in an abstract number such as matching costs), but explicitly described in the form of underfill and overfill errors.

The article is organized as follows. We first lay the groundwork for our contribution with an analysis of the AR performance evaluation problem, including a survey of selected publications from the past six years of AR research. This is followed by the introduction of AR event categories that extend Ward et al.'s [2006a] scoring system (Section 3). We then introduce a new system of frame and event metrics and show how they are applied (Section 4). The metrics are then evaluated by application to results from three previously published datasets (Section 5), followed by a concluding analysis of their benefits and limitations (Section 6).

2. PERFORMANCE EVALUATION

In its general form AR is a multiclass problem with c "interesting" classes plus a "NULL" class. The latter includes all parts of the signal where no relevant activity has taken place. In addition to insertions and deletions such multiclass problems can produce substitution errors that are instances of one class being mistaken for another. Note that insertions and deletions are a special case of a substitution with one of the classes involved being the NULL class.

In this article, we approach performance evaluation of multiclass AR by considering a class at a time. In doing so, the root problem we address is the characterization and summary of performance in a single, time-wise continuous, binary classification. That is, the output of the classifier at any one time is either positive, p or negative, n. Evaluation can then be viewed as a comparison of two discrete time-series (recognition output versus ground truth). We know that there is no objectively "best" similarity measure for time series comparison. The quality of the similarity measure depends on the application domain and the underlying assumptions. Here, we make two fundamental assumptions:

(1) Ground truth and classifier prediction are available for each individual frame of the signal.

(2) The time shift in which events are detected in the classifier output is at most within the range of the event. This means that events in the

recognition output can be assigned to events in the ground truth based on their time overlap. For example, assume that we have two events, e_1 and e_2, in the ground truth. If output r_x has temporal overlap with e_1, then we assume that it is a prediction for e_1 (similarly for e_2). If it has no temporal overlap with either of the two, then we assume it to be an insertion.[1] This allows us to do error scoring without having to worry about permutations of assignments of events from the ground truth to the classifier prediction. From our study of published work, we have found this assumption to be plausible for most applications.

2.1 Existing Methods for Error Scoring

Performance metrics are usually calculated in three steps. First, a comparison is made between the returned system output and what is known to have occurred (or an approximation of what occurred). From the comparison a scoring is made on the matches and errors. Finally, these scores are summarised by one or more metrics, usually expressed as a normalised rate or percentage.

Two basic units of comparison are typically used—frames or events:

Scoring Frames. A *frame* is a fixed-length, fixed-rate unit of time. It is often the smallest unit of measure defined by the system (the sample rate) and in such cases approximates continuous time. Because of the one-to-one mapping between ground and output, scoring frames is trivial, with frames assigned to one of: true positive (TP), true negative (TN), false positive (FP) or false negative (FN).

Scoring Events. We define an *event* as a variable duration sequence of positive frames within a continuous time-series. It has a start time and a stop time. Given a test sequence of g known events, $E = \{e_1, e_2, \ldots, e_g\}$, a recognition outputs h return events, $R = \{r_1, r_2, \ldots, r_h\}$. There is not necessarily a one-to-one relation between E and R. A comparison can instead be made using alternative means: for example DTW [Berndt and Clifford 1994], measuring the longest common subsequence [Agrawal et al. 1995], or a combination of different transformations [Perng et al. 2000]. An event can then be scored as either correctly detected (C); falsely inserted (I'), where there is no corresponding event in the ground truth; or deleted (D), where there is a failure to detect an event.

Commonly recommended frame-based metrics include: true positive rate ($tpr = \frac{TP}{TP+FN}$), false positive rate ($fpr = \frac{FP}{TN+FP}$), precision ($pr = \frac{TP}{TP+FP}$); or some combination of these (see 4.1.1). Similarly, event scores can be summarized by precision ($\frac{correct}{output\ returns}$), recall ($\frac{correct}{total}$), or simply a count of I' and D.

2.2 Shortcomings of Conventional Performance Characterization

Existing metrics often fall short of providing sufficient insight into the performance of an AR recognition system. We illustrate this using the examples in Figure 1. These plot a short section (300 s) of results described by Bulling et al.

[1]Note that it is permissible for r_x to overlap with both part of e_1 and part of e_2 (and possibly more events). Another permissible variant is that several events in the output overlap with one event in the ground truth.

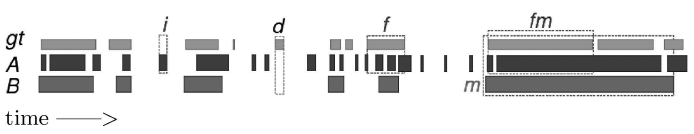

Fig. 1. Recognition results from a 300 s extract of the reading experiment reported by Bulling et al. [2008]. A sequence of 11 ground truth events (*gt*) are shown alongside outputs for unsmoothed (*A*) and smoothed (*B*) recognition. Five event errors are highlighted: (i) insertion, (d) deletion, (f) fragmentation, (m) merge, and (fm) fragmented and merged. For each sequence, the table shows the % event recall *Rec*. and % event precision *Pre*.; as well as the % frame-based true positive and false positive rates, *tpr* and *fpr*, and precision *pr*.

[2008] on the recognition of reading activities using body-worn sensors. Plot *A* shows a classifier output with classes "reading" versus "not reading"; plot *B* shows the same output but smoothed by a 30s sliding window; and *gt* shows the annotated ground truth. For both A and B, traditional frame metrics (*tpr*, *fpr*, *pr*) are calculated, as are event-based precision and recall (*Pre*., *Rec*.). For the event analysis, a decision needs to be made as to what constitutes a 'correct' event. Here, we define a true positive event as one that is detected by at least one output. We also decide that several events detected by a one output count only as a single true positive.

The frame results show that the *fpr* of A is almost 10% higher than that of B. Together with a poorer event precision, this indicates a larger number of false insertions in A. A's frame *tpr* is almost 10% lower than B. This might suggest more deletions, and thus a lower recall—but in fact its recall is 8% higher. Why? The answer is not clear from the metrics alone so we have to look at the plots. This instantly shows that A is more fragmented than B—many short false negatives break up some of the larger events. This has the effect of reducing the true positive frame count, while leaving the event count (based on the above assumption of 'detected at least once') unaffected.

Three key observations can be made of Figure 1: (1) some events in *A* are fragmented into several smaller chunks (f); (2) multiple events in *B* are recognized as a single merged output (m); and (3) outputs are often offset in time. These anomalies represent typical fragmenting, merge and time errors, none of which are captured by conventional metrics. Frame error scores of *false positive* or *false negative* simply do not distinguish between frames that belong to a "serious" error, such as insertion or deletion, and those that are timing offsets of otherwise correct events. Traditional event-based comparisons might be able to accommodate offsets using techniques such as dynamic time warping (DTW), or fuzzy event boundaries. However, they fail to explicitly account for fragmented or merged events.

2.3 Significance of the Problem

To assess the prevalence of fragmenting, merge and timing errors, we surveyed a selection of papers on continuous AR published between 2004 and 2010 at selected computing conferences and journals (e.g., Pervasive, Ubicomp, Wearable Computing, etc.) Table I highlights the main metrics used by each work, and whether these were based on frame, event, or some combination of both

Table I. The Metrics Used in a Selection of Continuous AR Papers

Reference	Frame			Event					Artefacts			
	Acc.	P,R	tpr, fpr	Conf.	Acc.	P,R	eDist.	I,D	EDD	Time	Frag.	Merge	
This work 2010		✓	✓			✓		✓		✓	✓	✓	
van Kasteren et al. 2010		f_1											
Maekawa et al. 2010		✓		✓									
Albinali et al. 2009	✓	✓								✓			
Zinnen et al. 2009						✓				✓			
Buettner et al. 2009						✓				✓	✓	✓	
Bulling et al. 2008			✓						✓	✓	✓	✓	
Choujaa and Dulay 2008		f_1			✓					✓			1
Huynh et al. 2008		✓											
Minnen et al. 2007	✓								✓	✓	✓		
Logan et al. 2007	✓		*AUC*										
Huynh et al. 2007	✓	✓											
Stiefmeier et al. 2006									✓	✓	✓	✓	2
Ward et al. 2006b		✓						✓	✓	✓	✓	✓	
Fogarty et al. 2006						f_1				✓	✓		
Amft and Troester 2006						✓		✓					
Lester et al. 2005	✓	✓								✓	✓	✓	
Patterson et al. 2005	✓						✓			✓	✓		
Westeyn et al. 2005					✓			✓		✓			
Lukowicz et al. 2004					✓			✓		✓			3
Bao and Intille 2004	✓									✓			
Tapia et al. 2004	✓				✓					✓	✓		4

Notes: 1) Defines a correct event as, 'occurred at least once during the day'.
2) Uses separate frame and event rates based on I, D, M, F, O & U.
3) Events counted by hand.
4) Scores three categories: percentage of activity duration correctly detected; event detected within interval; and event detected at least once.

Frame metrics include: accuracy (Acc.); precision and recall (P,R)—which are sometimes combined as $f_1 = 2 \cdot (P \cdot R)/(P + R)$; true and false positive rates (tpr, fpr)—or area under curve of tpr against fpr (AUC); the event confusion matrix (Conf); Acc. and P,R are also used as event metrics, as is edit distance (eDist.), and insertion and deletion counts (I,D). Error division diagram (EDD) is a hybrid frame-event method of presenting results. Also indicated are papers that, either through example plots, or through explicit discussion, exhibit artefacts of timing mismatch, fragmenting or merge.

evaluation methods. The final three columns indicate, either through explicit mention in the article, or through evidence in an included graph, whether artefacts such as timing errors, fragmenting or merge were encountered.

The simple frame-based accuracy metric was heavily used in earlier work (often accompanied by a full confusion matrix), but has since given way to the pairing of precision and recall. Event analysis has been applied by several researchers, however there is no clear consensus on the definition of a "correct" event, nor on the metrics that should be used. In most, however, there is strong evidence of timing offsets being an issue. Several highlight fragmenting and merge (though only those using EDD acknowledge these as specific error categories).

3. EXTENDED METHODS USING ADDITIONAL ERROR CATEGORIES

Ward et al. [2006a] introduced an extension to the standard frame scoring scheme that we adopt here for the single class problem. First, we introduce

additional categories of events to capture information on fragmenting and merge behavior. We then show how these are scored in an objective and unambiguous way.

3.1 Addition Event Information

In addition to insertions I' and deletions D, we define three new event categories:

Fragmentation. This is when an event in the ground truth is recognised by several returns in the output. An example of this is shown in Figure 1, where the event marked f is returned as 4 smaller events by output A. We refer to such an event as *fragmented* (F) and the returned events that cause it as *fragmenting* returns (F').

Merge. This is when several ground truth events are recognised as a single return (the inverse of fragmentation). This is exemplified in Figure 1 by the long return in B (marked m) that covers 3 ground truth events. We say that these ground events are *merged* (M), and refer to the single return event as a *merging* return (M').

Fragmented and Merged. A ground event can be both fragmented and merged. Consider the section of A output that is marked fm in Figure 1. The first ground truth event is clearly fragmented (into two returns). But the second return in A also covers another event, thus merging the two. We refer to the first ground event as being both *fragmented and merged* (FM). Similarly, a returned event can be both *fragmenting and merging* (FM'). The long return in A that covers the two ground events is an example of this.

3.2 Scoring Segments

An alternative scoring strategy was introduced by Ward et al. [2006a] that provides a mid-way solution between the one-to-one mapping of frame scoring, while retaining useful information from event scoring. This hybrid scheme is based on the notion of *segments*. A segment is the largest part of an event on which the comparison between the ground truth and the output of the recognition system can be made in an unambiguous way. Segments are derived by comparing the system output with ground truth: any change in either the output or the ground truth marks a segment boundary. Unlike events, segments have a one-to-one relationship between the output and ground truth. For a binary problem, positive (p) versus negative (n), there are four possible outcomes to be scored: TP_s, TN_s, FP_s and FN_s. The false positive and false negative errors, FP_s and FN_s, can be divided into the following subcategories to better capture useful event information (the example in Figure 2(a) shows how these might be assigned):

Insertion, I_s. A FP_s that corresponds exactly to an inserted return, I.

Merge, M_s. A FP_s that occurs between two TP_s segments within a merge return (i.e., the bit that joins two events).

Overfill, O_s. A FP_s that occurs at the start (O_s^α) or end (O_s^ω) of a partially matched return, that is, the bit of a return that 'spills' over the beginning or end of a ground event. (Combined overfill $O = O^\alpha + O^\omega$.)

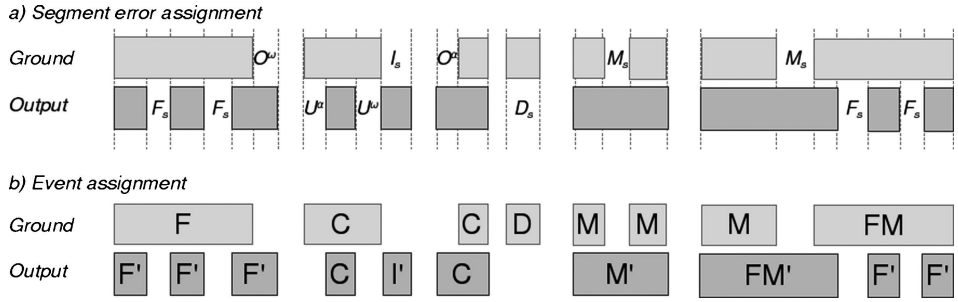

Fig. 2. Typical event anomalies found when comparing a ground truth with a (mock) recognition output: (a) shows the sequence divided into segments, with the FP_s and FN_s segments annotated as described in 3.2; (b) shows the same sequence with all of its ground and output events annotated with the event scores described in 3.4.

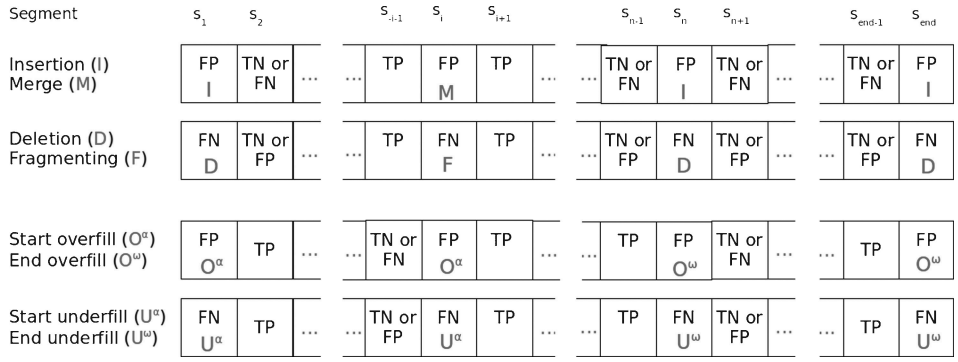

Fig. 3. Assignment of segment error types (based on prior assignment of FP_s, FN_s, TP_s and TN_s). All possible error assignments are shown here for the start s_1, middle (s_i and s_n) and end s_{end} of a sequence. For example, an FP_s segment at the start of a sequence that directly occurs before a FN_s or TN_s segment (s_1 on the top row) is classed as an insertion (I). An FN_s that occurs between two TP_s (e.g., s_i on the 2nd row) would be classed as fragmenting (F).

Deletion, D_s. A FN_s that corresponds exactly to a deleted event, D.

Fragmenting, F_s. A FN_s that occurs between two TP_s segments within a fragmented event (i.e., the "bad" fragment).

Underfill, U_s. A FN_s that occurs at the start (U_s^α) or end (U_s^ω) of a detected event, that is, the timing offset that effectively "deletes" part of the beginning or end of an event. (Combined underfill $U = U^\alpha + U^\omega$.)

Segments are scored according to the following procedure: First, assign every segment to one of the four standard scores. In a second pass, the FP_s and FN_s scored segments are further assigned to one of the 8 new error categories. This is done for each segment s_i at index i by considering the preceding s_{i-1} and following s_{i+1} segments. Figure 3 shows how assignments are made for every possible combination of s_{i-1}, s_i and s_{i+1}. Assignments are also shown for segments at the very beginning or end of a sequence (s_1 or s_{end}). As an example, the sequence TP_s-FP_s-TP_s would classify s_i as Merge. Alternatively, FN_s-FP_s-TP_s would yield s_i as a starting Overfill.

3.3 Scoring Frames

Once we have assigned error categories to segments, it is a simple matter to transfer those assignments to the frames that constitute each segment. Thus, we have counts of frame insertions I_f, deletions D_f, merge M_f, fragmenting F_f, overfill (O_f^α, O_f^ω) and underfill (U_f^α, U_f^ω). We use these numbers in our subsequent frame analysis.

3.4 Deriving Event Scores Using Segments

Figure 2(b) shows an example of how event scores can be unambiguously assigned using information provided by the corresponding segment scores. Trivially, $I' \equiv I_s$ and $D \equiv D_s$. We can also assign F ("fragmented event") to any event that contains at least one instance of an F_s segment. Likewise we assign M' ("merging return") to any return that contains at least one instance of an M_s segment.[2]

A merged event, M, is then assigned to any event that overlaps in time with a merging return M'. Similarly, a fragmenting return, F', is assigned to any output event that overlaps with a fragmented event F. If an event is assigned both M and F, we call it a FM ("fragmented and merged"); similarly any return that is M' and F' is called FM' ("fragmenting and merging").

Note that a key difference between the frame (and segment) error scores and the event scores is that the former analysis focuses on characterising and reporting frame errors (FP and FN), whereas here we report on counts of *matched* events. Thus frame merge errors, represented by mr, are calculated from the number of false positive frames—the spaces in between ground truth events; whereas event merging, as introduced here, relates to the matched events M that have been merged.

Segments can also be used to define the "correct" event score (C). However, this requires assumptions being made as to what constitutes a correct event. One common assumption is that an event is correct if it contains at least one TP segment. This is a troublesome definition because it completely ignores the possibility of fragmentation. Such a measure might better be termed "occurred at least once", for example, as in Tapia et al. [2004] and Choujaa and Dulay [2008]. We assume that it is better to classify correct only those events that cannot be applied to any of the other event categories. A correct event as used here is one that is matched with exactly one return event.[3]

3.5 Limits of Time Shift Tolerance

A key concerns behind our work is to distinguish between errors that are caused by small shifts in the recognition timing (which may be irrelevant for many applications) and the more "serious" errors of misclassified instances. Unlike other methods, such as DTW, we do not attempt to mask timing related

[2]In the initial paper by Ward et al. [2006a], only I, D, F and M' were explicitly recorded (F', M, FM, FM' and "correct" C were ignored.) This missing information made it difficult previously to devise a complete visualisation of the event scores.

[3]We allow timing errors—thus a correct event can overfill or be underfilled.

errors but make them explicit. Thus, a recognition system that works well except for the fact that it produces events that are slightly shifted with respect to the ground truth will not appear to be artificially producing false positives or false negatives. Instead it will be described as being good in terms of spotting events, but with a timing problem.

This may seem surprising given the fact that our evaluation works on sequential segment comparison. The explanation stems from the fact that we do not work on segments statically defined from the ground truth. Instead segment definitions are derived from the relation between ground truth and recognition system output. Also the score for a segment is influenced by neighbouring segments. So long as the recognized event has an overlap with the ground truth there will be a segment that is identified as correct, and adjoining segments will be labelled as timing errors (or fragmentation/merge when relevant).

This explanation also exposes the limits of the time shift tolerance which are given by event duration. If the time shift of the recognition output is larger than event length then timing error becomes an insertion or deletion error (as there is no segment that is labelled as correct). Clearly, in cases that involve very short (in terms of the time scale of the sensor and recognition system), widely spaced events, this would be a problem. However, in AR, such cases are rare. Even simple gestures such as pushing a button on pulling a drawer open take in a range of a second which amounts to 30 frames of video or 50 to 100 frames at a typical accelerometer sampling rate. Moreover, many applications look at complex longer term activities that can tale many seconds or even minutes.

4. METRICS

Once we have compared the recognition output with its ground truth and calculated scores for both frames and events, we then need to define metrics for summarizing and presenting the results.

4.1 Frame Metrics

4.1.1 *Standard Metrics.* Accuracy ($\frac{TP+TN}{P+N}$) is the most commonly used metric that can be calculated from a confusion matrix. Its main drawback is that it hides information on the specific nature of errors (the proportions of FP and FN). Precision and recall avoid this problem and are well known in AR. They are useful when it is difficult to gauge the size of N [van Rijsbergen 1979]. One drawback of precision is that it is heavily affected by changes in the proportions of classes in the dataset (class skew) [Fawcett 2004]. For this reason we prefer the skew-invariant fpr metric paired alongside tpr. This pairing can be plotted as an ROC curve for parameter-independent evaluation [Provost et al. 1998]. This is sometimes summarised in a single area-under-curve (AUC) metric [Ling et al. 2003].

4.1.2 *2SET Metrics.* We extend the standard confusion matrix to include eight new error categories. This 2-class segment error table (2SET) is shown in Figure 4(a). In (b), we define eight new metrics based on these categories. In previous work, metrics were calculated as a percentage of the total experiment

a)

	p	n
p'	TP	I_f M_f O_f^α O_f^ω
n'	D_f F_f U_f^α U_f^ω	TN

b)

deletion (dr)	$= D_f/P$	insertion (ir)	$= I_f/N$
fragmenting (fr)	$= F_f/P$	merge (mr)	$= M_f/N$
start underfill (u^α)	$= U_f^\alpha/P$	start overfill (o^α)	$= O_f^\alpha/N$
end underfill (u^ω)	$= U_f^\omega/P$	end overfill (o^ω)	$= O_f^\omega/N$

$P = D_f + F_f + U_f^\alpha + U_f^\omega + \mathrm{TP}$ $\quad N = I_f + M_f + O_f^\alpha + O_f^\omega + \mathrm{TN}$

Fig. 4. (a) 2-class segment error table (2SET): columns p and n denote ground truth, rows p' and n' denote classifier returns. Derived frame rate metrics are shown in (b).

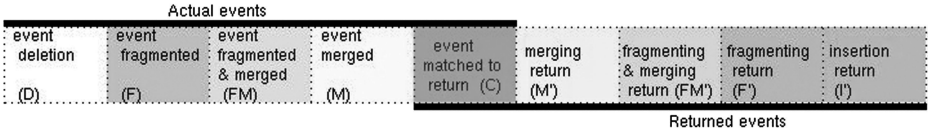

Fig. 5. Format of an event analysis diagram (EAD). A ground truth event can be assigned to exactly one of five categories: D, F, FM, M or correctly matched with exactly one returned event (C). Similarly, a returned event can be assigned to one of: C, M', FM', F' or I'.

time $N + P$. This formed the basis of the error division diagram (EDD). The problem with this is that any skew in the proportion of classes represented in a dataset can lead to results that cannot easily be compared across datasets. To maintain class skew invariance, the new 2SET metrics introduced here are based around *tpr* and *fpr*: that is, FN errors are expressed as a ratio of the total positive frames, P; and the FP errors are expressed as a ratio of the total negative frames, N. This means we can express $(1 - tpr) = dr + fr + u^\alpha + u^\omega$ and $fpr = ir + mr + o^\alpha + o^\omega$.

4.2 Event Metrics

From the categories laid out in 3.4, there are eight different types of event error scores. Four of these can be applied to ground truth events: deletions (D), fragmented (F), fragmented and merged (FM) and merged (M). The remaining four are applicable to returned events: merging (M'), fragmenting and merging (FM'), fragmenting (F') and insertions (I'). Together with correct events (C), these scores can be visualised in a single figure (see Figure 5), which we term the event analysis diagram (EAD). The sum of events $D+F+FM+M+C$ completely contains all of the possible events in ground truth. Likewise, $C+M'+FM'+F'+I$ completely contains all of the returned events in a system output. The EAD trivially shows exact counts of the event categories. For ease of comparison across differently sized datasets, these counts can also be conveniently reduced to rates or percentages: of total events $|E|$, $\frac{D}{|E|}$, $\frac{F}{|E|}$, $\frac{FM}{|E|}$, $\frac{M}{|E|}$ and $\frac{C}{|E|}$; or of total returns $|R|$, $\frac{C}{|R|}$, $\frac{M'}{|R|}$, $\frac{FM'}{|R|}$, $\frac{F'}{|R|}$ and $\frac{I}{|R|}$.

4.3 Application to Reading Example

4.3.1 *Frame Results.* To get an idea of how these metrics would be used in practice, we apply them to the examples of Figure 1—which we show again

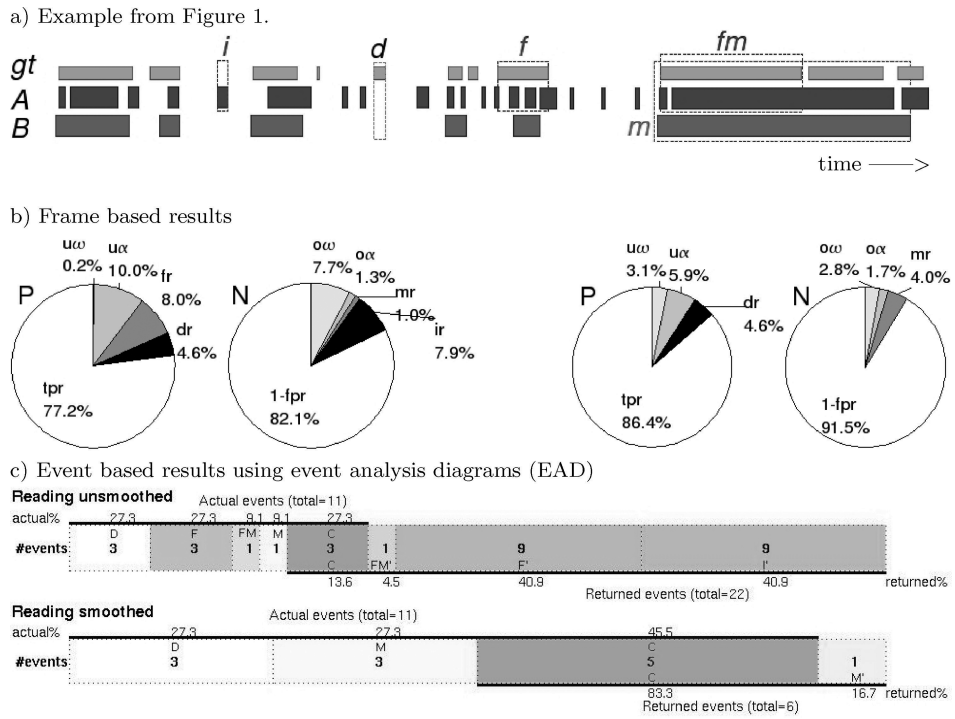

Fig. 6. Frame and event based analysis of Figure 1. Frame rates in (b) shown as a % of the total positive ground truth frames, P: tpr, dr, fr, u^{α} and u^{ω}. Rates shown as a % of the negative frames, N: true negative $(1 - fpr)$, ir, mr, o^{α} and o^{ω}. (The reading activity, P, takes up 47.6% of the 300s example data.) Note how the unsmoothed example contains fragmenting and insertion frames, whereas smoothed does not. The EADs of (c) show the number of actual (ground truth) events and returned (output) events for each event category (see Figure 5 for definitions). Also shown are the rates as % of the total actual events and as % of the total returned events.

in Figure 6(a). The frame results for the two examples, A (unsmoothed) and B (smoothed), are shown in pie chart format in Figure 6(b). For each result, one pie represents the breakdown of P frames ($tpr, dr, fr, u^{\alpha}, u^{\omega}$), the other of N frames ($1 - fpr, ir, mr, o^{\alpha}, o^{\omega}$).

At first glance, these figures reveal the most striking (visual) differences between the two examples: the existence of insertion (ir) and fragmenting (fr) errors in A, where none are seen in B. We also see that the fpr of A is greatly influenced by end overfill frames ($o^{\omega} = 7.7\%$ compared to the $o^{\omega} = 2.8\%$ in B). Start underfill is also higher in A ($u^{\alpha} = 10\%$) than B ($u^{\alpha} = 5.9\%$). This would indicate that the outputs in B are generally shifted later in time. This influence of inexact timing is not apparent when the standard metrics in Figure 1 are used.

The charts are useful indicators at explaining how much of the false negative and false positive frames are given over to specific types of error. However, they do not give any information on the distribution of these frames: for example, the high insertion rate ($ir = 7.9\%$) in A might be caused by many short insertions, or it might be a single long one. This is where an event analysis is useful.

4.3.2 *Event Results.* Figure 6(c) shows the EADs for each of the results. Instantly we see that A has many more returns than B (22 versus 6): with over 80% of A returns either fragmenting or insertions. In contrast 5 (83%) of B returns are correct, the remaining one is a single merge. Interestingly, this merge M' corresponds to exactly three merged events M in the ground truth (this is the m example from Figure 6(a)). This relationship between output and ground is typical for both merge and fragmenting. A fragmented event, for example, is reported by two complementary components: the number of fragmented events (F), and the number of correctly matching returned fragments (F'). Together, these paint a picture of how the output might look: In A, for example, we see three events that are fragmented (F) by a total of nine fragments (F').

A slightly more complicated relationship is represented by the fragmenting and merge output FM' in A: this corresponds to both the FM event the single M event and corresponds to the fm marked example in Figure 6(a).

With this detailed description of event performance, the EAD complements the frame analysis with information that would otherwise only be available using a visual analysis of the output plots.

5. DATASETS

To assess the utility of the proposed method, we use results calculated from three publicly available datasets: D1, from Bulling et al. [2008], D2, from Huynh et al. [2008], and D3, from Logan et al. [2007]. Following from the original papers, each set is evaluated using a different classifier: D1 using string matching; D2 using HMMs; and D3, decision tree. The aim of this diverse selection is to show that the method can be applied to a range of different datasets and using different classifiers. We do not intend to compare these results with one another (nor with the original results as published). Rather, we wish to show how results compare when presented using traditional metrics against those presented using our proposed metrics.

5.1 EOG Reading Dataset (D1)

The example in Figure 1 was taken from a study by Bulling et al. [2008] on recognizing reading activity from patterns of horizontal electrooculogram-based (EOG) eye movements. Six hours of data was collected using eight participants.[4] The activities in this dataset are very fine-grained. There are 706 distinct reading events, with event time-spans ranging from a few seconds up to several minutes at a time. Following the method described in the original paper, we use string matching on discretised sequences of horizontal eye movements. A threshold is applied to the output distance vector to determine "reading" or not. The output is then smoothed using a 30s majority vote sliding window.

5.2 Darmstadt Daily Routines Dataset (D2)

Huynh et al. introduced a novel approach for modelling daily routines using data from pocket and wrist-mounted accelerometers. They collected a 7-day,

[4]Download D1 at: http://www.andreas-bulling.de/publications/conferences/.

Table II. Frame and Event Results for D1, D2 and D3 using Standard Metrics

	Class	%	*tpr*	*fpr*	*pr.*	$\|E\|$	$\|R\|$	Rec.	Pre.
D1	Smoothed read	46.1	73.8	12.3	*83.7*	706	289	27.6	75.9
	Dinner	26.2	48.8	7.8	*22.1*	7	10	71.4	50
D2	Commuting	5.7	34.4	4.1	*34.4*	14	16	78.6	68.8
	Lunch	7.7	93.2	3.5	*68.8*	7	7	100	100
	Office work	56.1	89.8	10.5	*91.6*	27	17	55.6	100
	watch T.V.	12.6	62.2	4.4	*67.6*	18	380	100	6.3
	dishwashing	0.4	96.0	6.5	*5.9*	25	264	88	8.3
D3	eating	7.9	65.1	19.2	*22.4*	165	1111	66.7	10.8
	computer	32.4	78.2	18.5	*67.0*	77	820	93.5	18.7
	phone	4.3	54.4	24.6	*9.1*	97	1466	84.5	5.8

% of frames in dataset for each class shown alongside frame-based true positive rate, or recall (*tpr*), false positive rate (*fpr*) and precision (*pr.*) as %; and total ground events $|E|$, output events $|R|$, event recall (Rec.) and event precision (Pre.).

single-subject dataset. The dataset is comprised of a 13-dimension feature space; this includes the mean and variance of the 3-axis acceleration from the wrist and pocket sensors plus a vector based on time of day.[5] For simplicity, we replicate the method they use to provide a baseline in one of the several described experiments. We use Hidden Markov Models (HMM) to recognise each of four annotated routines (dinner, lunch, commuting and office work). A remaining 25% of the dataset is not modelled here (the unclassified case, or 'NULL'). We build a five state, left-to-right, mixed Gaussian HMM for each class using leave-one-day-out training. Each observation feature vector is modelled using a mixture of two Gaussian. The competing models are successively applied to a 30s sliding window. The highest likelihood model is chosen as the output class for each window.

5.3 MIT PLCouple1 Dataset (D3)

Logan et al. [2007] presented a study aimed at recognising common activities in an indoor setting using a large variety and number of ambient sensors. A single subject was tracked and annotated for 100 hours using the MIT PlaceLab [Intille et al. 2006]. A wide range of activities are targeted, five of which we choose as a representative sample of the dataset: watching T.V., dishwashing, eating, using a computer and using the phone.[6] The activities in this dataset are finer-grained than those of D2, covering relatively short durations (up to an hour) and over many more instances (between 18 and 165). The dataset also includes an example set of pre-computed output predictions calculated using a decision-tree classifier (on the MITES motion sensor data). It is these results that we use here.[7]

5.4 Application of Metrics to Datasets

5.4.1 *Standard Frame and Event Analysis.* Table II shows how the results from the three datasets might be analysed using standard metrics. Judging by

[5]Download D2 at: http://www.mis.informatik.tu-darmstadt.de/data.
[6]Results for the remaining activities are available on request.
[7]Download D3 at: http://architecture.mit.edu/house_n/data/PlaceLab/PLCouple1.htm.

tpr and fpr, most classes (with the exception of "dinner" and "commuting" in D2; and "phone" in D3), seem to be recognised fairly well (*tpr* above 60%, fpr below 20%). These numbers are misleading, however. The event metrics reveal very different results. Most notably, all of the classes in D3 suffer from very low event precision (between 6% and 19%) And reading in D1 has a recall of less than 30%. On the plus side, it reveals that "Lunch" in D2 is recognised with perfect event accuracy. This simple comparison already shows the importance of considering both frame and event analysis when presenting recognition results in AR.

5.4.2 *Using 2SET Frame Metrics.* We extend the interpretation further by analysing the specific frame errors in the pie chart pairings of Figure 7. The "P" charts clearly show how the poor frame *tpr* results for "dinner", "commuting" and "phone" are comprised. Alongside the deletion frame errors (*dr*) for these classes, we see that many of the frames have been underfilled (u^{α} and u^{ω}). This indicates something that is not visible from the standard metrics: that timing offsets often constitute a large portion of what is regarded as frame error. The opposite is also shown here: the "N" charts for the D3 classes show that by far the most common frame errors within fpr are insertions (*ir*). High *ir* correlates with what might be expected given the low event precision for these classes.

5.4.3 *Using EAD Event Metrics.* Finally, we flesh out the event-based results using the EADs of Figure 8. This reveals a number of useful findings, including:

Reading, D1. Over half of reading events are merged together (M = 373 merged events). These merges are caused by 96 separate merging outputs (M').

Dinner and Commuting, D2. The event results for both of these activities correlate well to the standard analysis in Table II—they include only deletion and insertion errors.

Office Work, D2. Almost 52% (14) of these events are merged together into 6 large merge outputs. Two of the events are also fragmented. None of these characteristics are apparent in the standard evaluation.

Computer, D3. Not shown by the standard metrics, most of the output returns for computer (58.9%) are fragmenting. (Fragmenting also plays a large part in the result for watch TV, albeit to a lesser extent at 28%).

6. DISCUSSION

6.1 Highlighting the Benefits

To illustrate the benefits of the proposed metrics we take a second, more detailed look at two examples from the data presented in 5.4.

Frame Level D2 Dinner and D3 Phone. In both classes, around 50% of the positive frames are correctly recognized. Using traditional metrics, this would imply around half of the correct frames being deleted or a recall of 0.5 which, by all standards, is a poor performance. Thus, in both cases, an application designer may by inclined not to use the system, or find a work-around that

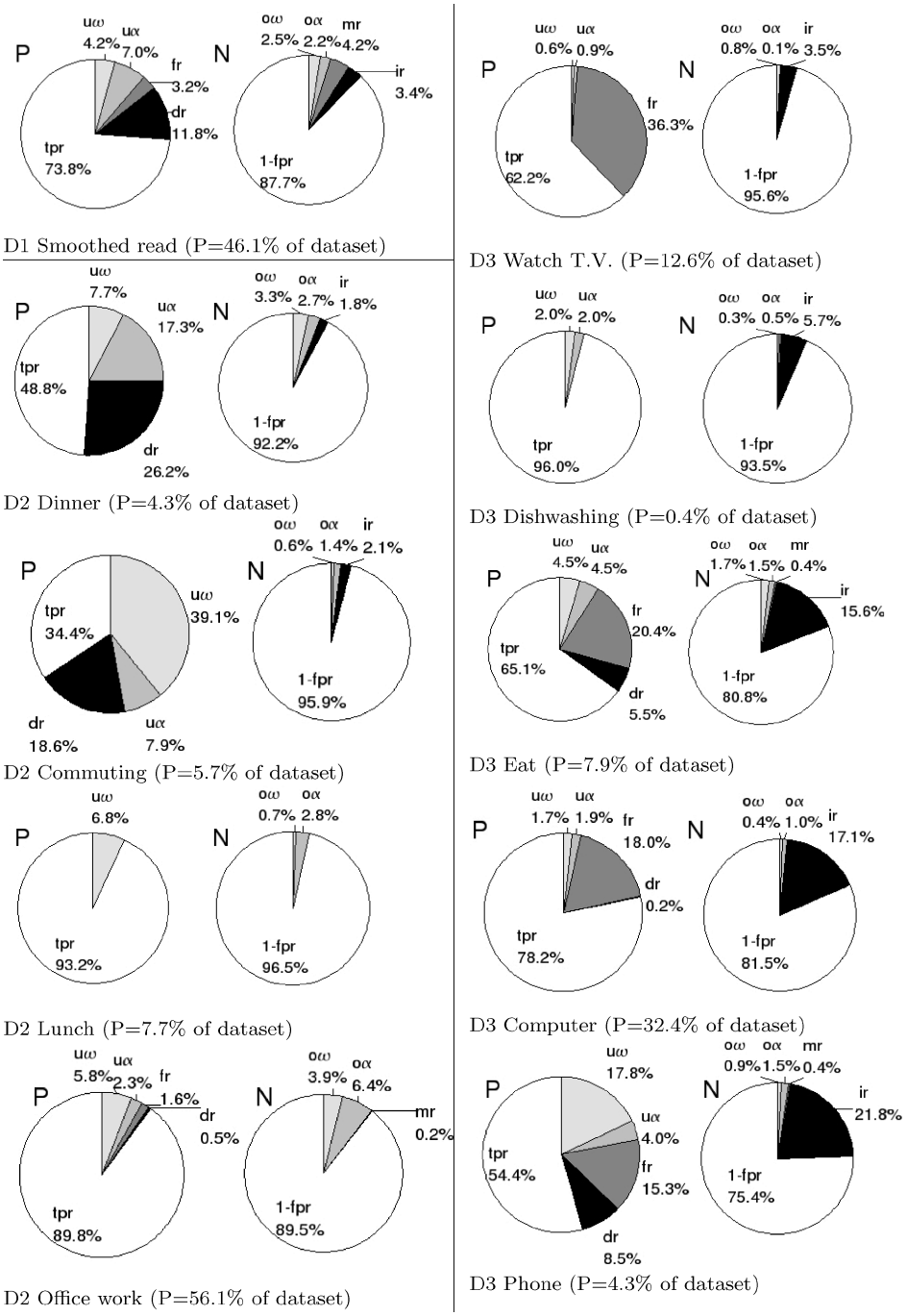

Fig. 7. Frame-based error results for each class of the 3 datasets, D1, D2, and D3. Pie chart pairs represent error rates as percentages of the total positive ground truth frames, P and of the total negative frames, N. See Figure 4 for definitions of metrics used.

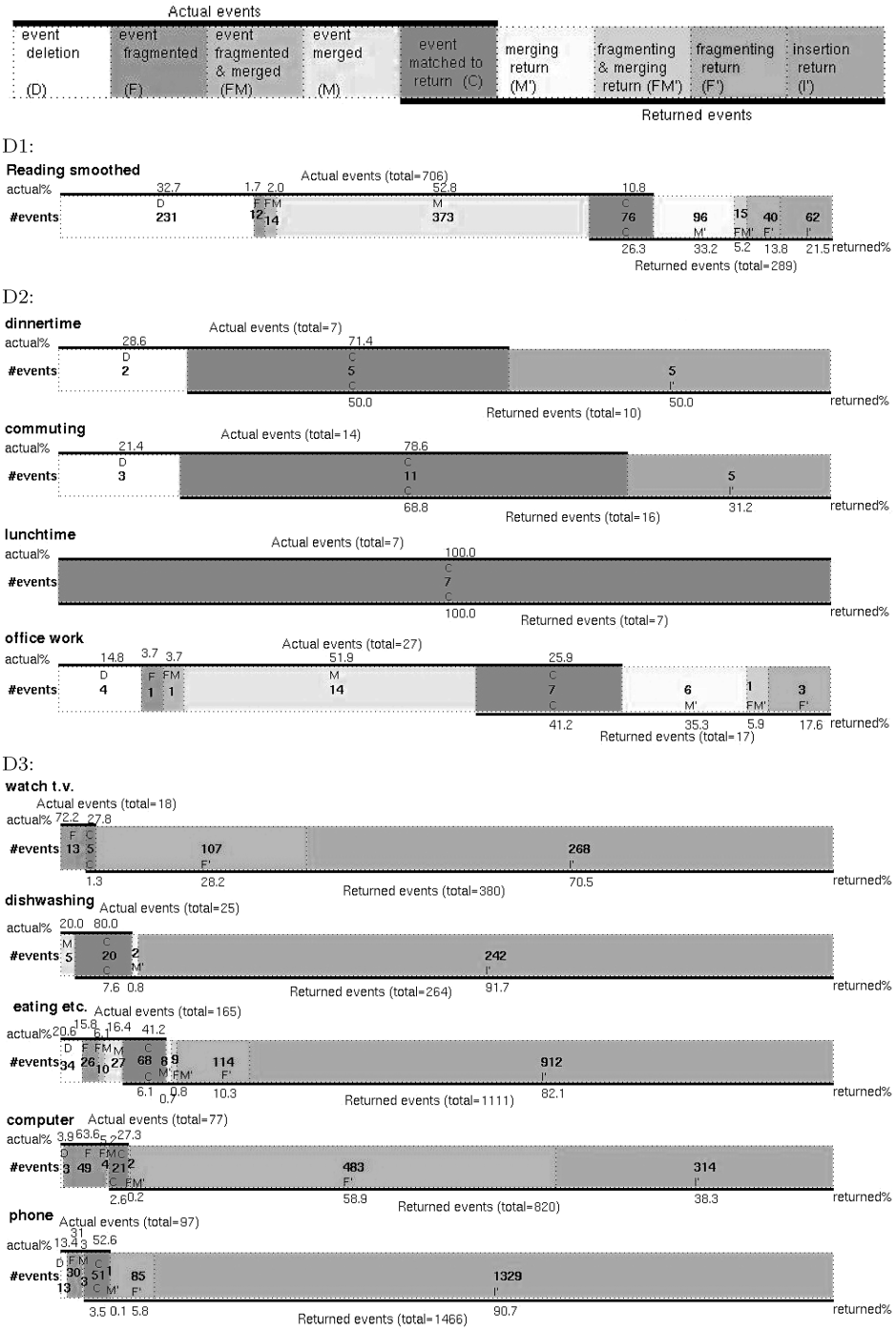

Fig. 8. Event summary for each class in the 3 datasets, with key to categories shown at top.

does not require the recognition of the particular classes. However, looking at the corresponding chart in Figure 7 we see that the extended metrics provide a different picture. For both classes, around half of non-recognized true positive frames are due to timing errors, not real deletions. From an application point of view, this is already a significantly better picture (unless timing is crucial for the application because it for example measures the time spent during different activities). For the dinner class, the remaining quarter are frame errors caused by real deletions. However, for the phone class, there are only around 8% real deletion related frames and 15% fragmentation related frame errors (frames which have been assigned to a different class because at the specific location the event was fragmented). Thus, if an application does not care about fragmentations, then the effective rate of deletion related frames is just 8% instead of 50% as suggested by traditional metrics.

D3 Watching TV, Dishwashing and Computer. All three classes have poor event level precision (6.3% for TV, 8.3% for dishwashing , and 18.7% for computer). This implies that the number of events that the system has returned is between 5 (for computer) and nearly 20 (for watching TV) times higher then the true number of events. In particular, for the watch TV class, this may again lead an application designer to discard the system as useless.

Taking into account traditional frame based analysis paints a different picture. The false positive frame rates are 4.4% for TV, 6.5% for the dishwasher and 18.5% for computer. Thus, only a small percentage of the frames are wrongly labelled as belonging to the respective class and one may think that we have a usable system. Also TV that was by far the poorest performing class on event level performs best on frame level. For the computer class, it is exactly the other way around.

Using traditional metrics, an application developer would get conflicting information from the frame and the event level analysis. An experienced developer may be able to make an educated guess and hypothesize that the discrepancy comes from timing and possible fragmentation issues. However, it is only by looking at the EAD and the 2SET charts generated by our approach that the full, consistent, and reliable picture emerges upon which an informed design decision can be made.

First, the EAD for dishwashing shows that the low event precision is almost entirely due to insertions which means that system performance is really as bad as traditional metrics suggest. The extended frame by frame metrics in Figure 7 confirms this as virtually all false positive frames are insertions (there are nearly no timing errors). Because there are many inserted events but the proportion of false positive frames is relatively small we can assume that the inserted events tend to be short. Taken together it tells the designer that (1) the systems often wrongly identifies events, (2) however, inserted events tend to be short, and (3) correctly spotted events are spotted very accurately in terms of timing and event length. This is far more information then can be extracted from the traditional metrics.

For the computer class, we can see that 58.9% percent of the returned events are fragmentations, while only 38.3% are insertions. This is less then half of

the number of insertions suggested by the traditional event recall metrics. For the application designer, it suggest that the class is not as bad as originally thought (assuming fragmentations are not a big issue). On frame level, our method reveals that most false positive frames are actually insertions while most false negative frames are due to fragmentations. There are virtually no overfill errors and the proportion of false positive frames is much higher (in particular considering the lower level of event insertions). Thus, we know that inserted events tend to be long. We also know that the system is very sensitive to event boundaries. For the designer of the recognition system, the combination of large number of fragmentation with high sensitivity to boundaries suggest a possible improvement: increasing the threshold for the recognition of event end. Clearly, this may reduce the number of fragmentations (since they are caused by the system mistakenly thinking that an event ends). It may, of course, increase the amount of overfill, but we know that so far there were nearly none so that may be acceptable.

For the TV class, the considerations are similar to the computer except that the impact of fragmentations is much lower (just 28% of the returned events). From the extended frame metrics, we can conclude that the insertions tend to be very short, while in the fragmentation events the "interruptions" are quite long. Neither overfill nor underfill is an issue.

In summary, the above examples clearly show that the metrics proposed in this article can provide useful information to both application designers who use a recognition system and the developer of the recognition system.

6.2 Limitations and Challenges for Future Work

6.2.1 *2SET Frame Metrics.* Timing errors are currently represented by rates based on start and end overfill and underfill ($o\alpha$, $o\omega$, $u\alpha$ and $u\omega$). Because these rates are calculated with respect to the total number of frames of positive ground truth (P) or the total frames of negative ground truth (N), this means that the effects of poor timing might get lost for datasets involving long activities. As an example, "Office work" reports a delayed detection of only $u\alpha = 5.8\%$. However, because this activity represents 56% of the dataset, the actual time involved with the delay may actually be quite large. Conversely, detection of the relatively short activity "Commuting" in D2 is typically delayed with $u\alpha = 39\%$. That's 39% of an activity that only takes up less than 6% of the total dataset. In real time, the actual delay in number of frames may be quite small (though for a short event this can be significant). One simple solution to this problem is to present the actual number of frames in addition to the rates.

6.2.2 *EAD Event Metrics.* The definition of "Correct" that we use in the analysis of events might be regarded as harsh, particularly for applications where, "correct if detected at least once," is preferred. EAD representation has the potential to render a poor correct count for results that might otherwise be regarded as quite acceptable. We believe that it is better to show all results in the brightest (coldest) light, and then give explanations afterwards if need be. Fragmented events, for example, might be acceptable for some applications. In

these cases, the events marked F may be aggregated with C and presented in an additional, application-specific metric.

6.2.3 *Extensions to Analysis.* A recommended practice is to consider performance over a range of operating points, such as using ROC [Provost et al. 1998]. One crude attempt to achieve this would be to sweep the proposed frame metrics as a series of stacked bar charts (rather than pie) alongside one another. For events, a series of EADs could be stacked on top of one another. An improved approach might be to aggregate a selection of some of the metrics and plot these in an ROC-style curve. How this might be done is a subject of ongoing research.

The metrics presented here can also be combined in a way that allows us to examine variance across different sets (e.g, participants, or classes). Again, a challenge for future work is how this information might be displayed in an informative way.

6.2.4 *Evaluating Multiple Classes.* The segment-based method presented by Ward et al. [2006a] is intrinsically multi-class: Each pairing of ground truth and output segment is assigned to exactly one of six categories (insertion-deletion, insertion-underfill, insertion-fragmenting, overfill-deletion, overfill-underfill, and merge-deletion). Scores of these errors are then recorded in a multi-class, confusion-matrix style *segment error table* (SET). Although SET completely captures both segment and frame errors, it can be difficult to interpret. But its main drawback is that there is no clear way of handling event errors—for example, in cases where an event of one class is fragmented by instances of several different classes.

We developed 2SET and EAD to work around these problems.[8] The assumption we make is that each activity class can be evaluated independently of all others—all else becomes, in effect, the NULL class. This assumption may not entirely hold for discriminative classifiers where the result for one class is influenced by the performance of the others. However, we believe the general practicability of our approach to outweigh this concern.

6.2.5 *Interpretation of Results.* With so many metrics to consider, it could be argued that the approach taken in this work does not lend itself well to a concise presentation of results in a research paper—particularly where many results and systems are to be compared. Researchers may choose instead to use a subselection of the most pertinent metrics to the specific problem being tackled. A single-value combination metric (similar to AUC or f_1) might be derived for optimization tasks. Exactly how this might be done is still the subject of ongoing research.

6.3 Related Work in Other Areas

The common approach to event analysis of AR has its roots in automatic speech recognition (ASR), adapting variants of the word error rate components, insertion, deletion and substitution, to the activity scenario [McCowan et al. 2004].

[8]Note that 2SET can be trivially derived from the full SET by collapsing inter-class substitutions.

But ASR, with its clearly defined atomic elements (words and characters) provides no framework for capturing the "difficult" issues of fragmenting, merge and vastly variable durations that occur in real world activity.

In early computer vision research, the problem of finding suitable ways of capturing difficult information was often sidestepped altogether in favor of showing typical example images [Hoover et al. 1996; Müller et al. 1999]. Though suitable for establishing the feasibility of a method with a small number of samples, this approach does not scale well for studies using large datasets. The development of a complete framework for performance remains an active area of research, particularly with video analysis [Kasturi et al. 2009; Zhang 2001].

Shafait et al. [2008] recently introduced a framework of scoring methods and metrics for image segmentation with the goal of characterizing performance where existing metrics fail. In particular, they highlighted "over" "under" and "missed" segmentation, which although expressed in a 2D context, are analogous to the temporal errors dealt with here.

7. CONCLUSION

We have shown that on results generated using published, nontrivial datasets, the proposed metrics reveal novel information about classifier performance, for both frame and event analysis. The additional information provided by fragmenting, merge, insertion, deletion, and timing errors allows crucial event information to be incorporated into the frame-by-frame evaluation. However, because it is based on total durations, or number of frames, this method of reporting can be misleading when activity event durations are variable. A single, long, correct event, for example, can mask the presence of multiple, shorter insertions (and vice-versa). Event-based evaluation gets around this problem where rates based on counts of correct and incorrect events are reported. However, AR researchers have largely avoided this method of evaluation, in part, because of the difficulty of scoring correct and incorrect activities. The introduction of a full characterization of fragmented and merged events, and a revised definition of insertions and deletions, provides one possible solution to these difficulties. We introduce the event analysis diagram (EAD) showing a complete breakdown of ground truth event counts alongside recognition output event counts. Rates based on these, together with timing information from a frame-analysis, can, we believe, provide a firm basis for a more complete evaluation of future work in activity recognition.

REFERENCES

AGRAWAL, R., LIN, K.-I., SAWHNEY, H. S., AND SHIM, K. 1995. Fast similarity search in the presence of noise, scaling, and translation in time-series databases. In *Proceedings of the 21st International Conference on Very Large Data Bases*. 490–501.

ALBINALI, F., GOODWIN, M. S., AND INTILLE, S. S. 2009. Recognizing stereotypical motor movements in the laboratory and classroom: a case study with children on the autism spectrum. In *Proceedings of the International Conference on Ubiquitous Computing*. ACM, New York, 71–80.

AMFT, O. AND TROESTER, G. 2006. Methods for detection and classification of normal swallowing from muscle activation and sound. In *Proceedings of the Pervasive Health Conference and Workshops*, 1–10.

BAO, L. AND INTILLE, S. 2004. Activity recognition from user-annotated acceleration data. In *Proceedings of the International Conference on Pervasive Computing*.

BERNDT, D. AND CLIFFORD, J. 1994. Using dynamic time warping to find patterns in time series. In *Proceedings of the Workshop on Knowledge Discovery in Databases (KDD)*. AAAI, 359–370.

BUETTNER, M., PRASAD, R., PHILIPOSE, M., AND WETHERALL, D. 2009. Recognizing daily activities with rfid-based sensors. In *Proceedings of the 11th International Conference on Ubiquitous Computing*. ACM, New York, 51–60.

BULLING, A., WARD, J. A., GELLERSEN, H.-W., AND TRÖSTER, G. 2008. Robust recognition of reading activity in transit using wearable electrooculography. In *Proceedings of the International Conference on Pervasive Computing*. 19–37.

CHOUJAA, D. AND DULAY, N. 2008. Tracme: Temporal activity recognition using mobile phone data. In *Proceedings of the IEEE/IFIP International Conference on Embedded and Ubiquitous Computing*. Vol. 1, 119–126.

FAWCETT, T. 2004. *ROC Graphs: Notes and Practical Considerations for Researchers*. Kluwer.

FOGARTY, J., AU, C., AND HUDSON, S. E. 2006. Sensing from the basement: a feasibility study of unobtrusive and low-cost home activity recognition. In *Proceedings of the 19th Symposium on User Interface Software and Technology*. ACM, New York, 91–100.

HOOVER, A., JEAN-BAPTISTE, G., JIANG, X., FLYNN, P., BUNKE, H., GOLDOF, D., BOWYER, K., EGGERT, D., FITZGIBBON, A., AND FISHER, R. 1996. An experimental comparison of range image segmentation algorithms. *IEEE Trans. Patt. Anal. Mach. Intell. 18*, 7, 673–689.

HUYNH, T., BLANK, U., AND SCHIELE, B. 2007. Scalable recognition of daily activities with wearable sensors. In *Proceedings of the Conference on Location- and Context-Awareness*.

HUYNH, T., FRITZ, M., AND SCHIELE, B. 2008. Discovery of activity patterns using topic models. In *Proceedings of the International Conference on Ubiquitous Computing*. ACM, New York, 10–19.

INTILLE, S. S., LARSON, K., TAPIA, E. M., BEAUDIN, J., KAUSHIK, P., NAWYN, J., AND ROCKINSON, R. 2006. Using a live-in laboratory for ubiquitous computing research. In *Proceedings of the International Conference on Pervasive Computing*.

KASTURI, R., GOLDGOF, D., SOUNDARARAJAN, P., MANOHAR, V., GAROFOLO, J., BOWERS, R., BOONSTRA, M., KORZHOVA, V., AND ZHANG, J. 2009. Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol. *IEEE Trans. Patt. Anal. Mach. Intell. 31,* 2, 319–336.

LESTER, J., CHOUDHURY, T., KERN, N., BORRIELLO, G., AND HANNAFORD, B. 2005. A hybrid discriminative/generative approach for modeling human activities. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. 766–772.

LING, C. X., HUANG, J., AND ZHANG, H. 2003. Auc: A statistically consistent and more discriminating measure than accuracy. In *Proceedings of the 18th International Conference on Artificial Intelligence (IJCAI)*. 329–341.

LOGAN, B., HEALEY, J., PHILIPOSE, M., MUNGUIA-TAPIA, E., AND INTILLE, S. 2007. A longitudinal evaluation of sensing modalities for activity recognition. In *Proceedings of the International Conference on Ubiquitous Computing*.

LUKOWICZ, P., WARD, J., JUNKER, H., TRÖSTER, G., ATRASH, A., AND STARNER, T. 2004. Recognizing workshop activity using body worn microphones and accelerometers. In *Proceedings of the International Conference on Pervasive Computing*.

MAEKAWA, T., YANAGISAWA, Y., KISHINO, Y., ISHIGURO, K., KAMEI, K., SAKURAI, Y., AND OKADOME, T. 2010. Object-based activity recognition with heterogeneous sensors on wrist. In *Proceedings of the International Conference on Pervasive Computing*. 246–264.

MCCOWAN, I., MOORE, D., DINES, J., GATICA-PEREZ, D., FLYNN, M., WELLNER, P., AND BOURLARD, H. 2004. On the use of information retrieval measures for speech recognition evaluation. IDIAP-RR 73, IDIAP, Martigny, Switzerland.

MINNEN, D., WESTEYN, T., ASHBROOK, D., PRESTI, P., AND STARNER, T. 2007. Recognizing soldier activities in the field. In *Body Sensor Networks*, vol. 13, 236–242.

MÜLLER, H., MÜLLER, W., SQUIRE, D. M., MARCHAND-MAILLET, S., AND PUN, T. 1999. Performance evaluation in content-based image retrieval: Overview and proposals. Tech. rep., University of Geneve, Switzerland.

PATTERSON, D., FOX, D., KAUTZ, H., AND PHILIPOSE, M. 2005. Fine-grained activity recognition by aggregating abstract object usage. In *Proceedings of the International Symposium on Wearable Computers*. IEEE, 44–51.

PERNG, C.-S., WANG, H., ZHANG, S., AND PARKER, D. 2000. Landmarks: a new model for similarity-based pattern querying in time series databases. In *Proceedings of the International Conference on Data Engineering*. IEEE, 33–42.

PROVOST, F., FAWCETT, T., AND KOHAVI, R. 1998. The case against accuracy estimation for comparing induction algorithms. In *Proceedings of the 15th International Conference on Machine Learning* 445–453.

SHAFAIT, F., KEYSERS, D., AND BREUEL, T. 2008. Performance evaluation and benchmarking of six-page segmentation algorithms. *IEEE Trans. Patt. Anal. Mach. Intell. 30,* 6, 941–954.

STIEFMEIER, T., OGRIS, G., JUNKER, H., LUKOWICZ, P., AND TROESTER, G. 2006. Combining motion sensors and ultrasonic hands tracking for continuous activity recognition in a maintenance scenario. In *Proceeding of the International Symposium on Wearable Computers*. 97–104.

TAPIA, E., INTILLE, S., AND LARSON, K. 2004. Activity recognition in the home using simple and ubiquitous sensors. In *Proceedings of the International Conference on Pervasive Computing*.

VAN KASTEREN, T., ENGLEBIENNE, G., AND KRÖSE, B. 2010. Transferring knowledge of activity recognition across sensor networks. In *Proceedings of the International Conference on Pervasive Computing*. 283–300.

VAN RIJSBERGEN, C. 1979. *Information Retrieval* 2nd Ed. Dept. of Computer Science, University of Glasgow.

WARD, J., LUKOWICZ, P., AND TRÖSTER, G. 2006a. Evaluating performance in continuous context recognition using event-driven error characterisation. In *Proceedings of the Symposium on Location and Context Awareness*. 239–255.

WARD, J., LUKOWICZ, P., TRÖSTER, G., AND STARNER, T. 2006b. Activity recognition of assembly tasks using body-worn microphones and accelerometers. *IEEE Trans. Patt. Anal. Mach. Intell. 28,* 10, 1553–1567.

WESTEYN, T., VADAS, K., BIAN, X., STARNER, T., AND ABOWD, G. D. 2005. Recognizing mimicked autistic self-stimulatory behaviors using HMMs. In *Proceedings of the International Symposium on Wearable Computers*. 164–169.

ZHANG, Y. 2001. A review of recent evaluation methods for image segmentation. In *Proceedings of the International Symposium on Signal Processing and its Applications*. 148–151.

ZINNEN, A., WOJEK, C., AND SCHIELE, B. 2009. Multi activity recognition based on bodymodel-derived primitives. In *Proceedings of the Symposium on Location- and Context-Awareness*. 1–18.

Received March 2010; June 2010; accepted July 2010