

The Effect of Sampling Rate on the Performance of Template-based Gesture Recognizers

Radu-Daniel Vatavu
University Stefan cel Mare of Suceava
13 Universitatii
Suceava, Romania
vatavu@eed.usv.ro

ABSTRACT

We investigate in this work the effect of motion sampling rate over recognition accuracy and execution time for current template-based gesture recognizers in order to provide performance guidelines to practitioners and designers of gesture-based interfaces. We show that as few as 6 sampling points are sufficient for Euclidean and angular recognizers to attain high recognition rates and that a linear relationship exists between sampling rate and number of gestures for the dynamic time warping technique. We report execution times obtained with our controlled downsampling which are 10–20 times faster than shown by existing work at the same high recognition rates. The results of this work will benefit practitioners by providing important performance aspects to consider when using template-based gesture recognizers.

Categories and Subject Descriptors

H.5.2 [[Information Interfaces and Presentation]]: User Interfaces; I.5.2 [[Pattern Recognition]]: Design Methodology—Classifier design and evaluation

General Terms

Algorithms, Experimentation, Performance

Keywords

gesture recognition, recognition rate, classifier performance, execution time, comparing classifiers, sampling resolution, experimentation

1. INTRODUCTION

Gesture-based commands have become increasingly accessible for user interfaces in the measure that sensing and recognition technologies developed to allow reliable acquisition and robust classification of users' input motions. Although under the appanage of pattern recognition experts, several recognizers have been proposed for or by the human-computer interaction (HCI) community in order to enable

practitioners and designers to rapidly prototype and experiment gesture-based interfaces [8, 10, 11, 14, 23].

Most of these recognizers use the principle of nearest neighbor classification (NN) which works by comparing the gesture candidate to a set of examples (referred to as the training set). A metric is used to compute the dissimilarity between two gesture executions describing *how far* they are positioned one from another in some feature space. The candidate is recognized as belonging to the class of its *closest* or *nearest* example.

Nearest-neighbor classifiers provide several advantages over other recognition techniques such as ease of implementation and clear understanding of the inner workings even for non-experts. Also, there is no need for complex parametric modeling that most pattern recognition techniques demand such as weights computation for neural networks [20, 22] or probabilities estimation for hidden markov models [1, 16]. Still, probably the most important advantages of the NN approach for HCI are its flexibility and adaptability: new or user-specific gestures can be added by simply providing training examples without the need of rethinking the structure or retraining the classifier. Such advantages have been remarked before [3, 11] and many nearest-neighbor classifiers have been proposed with high reported recognition rates [10, 11, 23]. While the search algorithm for the closest neighbor is always the same, the proposed classifiers differ in the metric they employ which decides their performance in practice.

The performance of a recognizer is primarily measured by its *recognition rate* which represents the estimate of the percentage of gestures which are expected to be correctly recognized. While the recognition rate represents the most important criterion, there are other factors that determine system performance such as the *execution time* required for classification and the amount of *system memory* employed for storing the training set. Their importance is especially revealed for mobile computing for which resources are restrictive.

The metrics that proved successful for NN gesture recognizers are the Euclidean distance [2, 8, 10, 23], dynamic time warping [15, 18], and the angular inverse cosine distance [9, 11]. They all work directly on the motion input that is normalized and resampled into the same number of n equally-distanced points [5, 10, 11, 23]. Although not needed for dynamic time warping (DTW), resampling the raw input gestures at the same rate is mandatory for computing the Euclidean and angular metrics that require point-to-point computations. Table 1 briefly describes the metrics and shows their time complexities. Note that these are met-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ICMI'11, November 14–18, 2011, Alicante, Spain.

Copyright 2011 ACM 978-1-4503-0641-6/11/11 ...\$10.00.

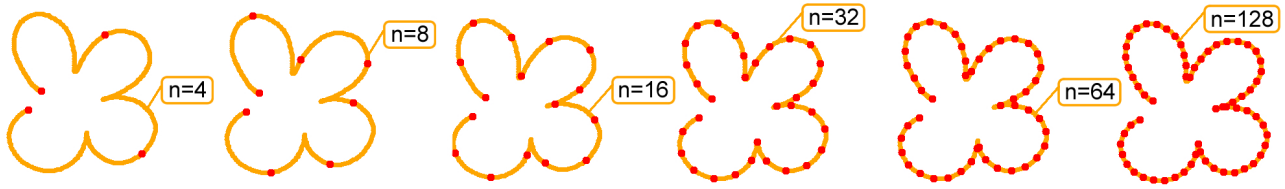


Figure 1: Sampling resolution rates n ranging from under- to over-sampling for the same gesture motion.

Metric	Definition	Time
Euclidean (d_1)	$d_1(p, q) = \sum_{i=0}^{n-1} \ p_i - q_i\ = \sum_{i=0}^{n-1} \sqrt{(p_{ix} - q_{ix})^2 + (p_{iy} - q_{iy})^2}$	$O(n)$
Dynamic time warping (d_2)	$d_2(p, q) = c_{n-1, n-1}$ $c_{i, j} = \ p_i - q_j\ + \min \begin{cases} c_{i-1, j-1} \\ c_{i-1, j} \\ c_{i, j-1} \end{cases}$ $i, j \in \{0..n-1\}$	$O(n^2)$
Angular inverse cosine (d_3)	$d_3(p, q) = \arccos \frac{p \cdot q}{\ p\ \cdot \ q\ }$ $p \cdot q = \sum_{i=0}^{n-1} (p_{ix}q_{ix} + p_{iy}q_{iy})$ $\ p\ = \sum_{i=0}^{n-1} \sqrt{p_{ix}^2 + p_{iy}^2}$	$O(n)$

Table 1: Gesture metrics commonly used with nearest-neighbor recognizers working directly on the motion inputs $p = \{p_i = (p_{ix}, p_{iy}) \in \mathbb{R}^2 | i = 0, n-1\}$ and $q = \{q_i = (q_{ix}, q_{iy}) \in \mathbb{R}^2 | i = 0, n-1\}$. Note: $\|a - b\|$ denotes the regular Euclidean distance between points a and b from \mathbb{R}^2 .

rics defined over the gesture space not to be confounded with local metrics between points (such as the regular Euclidean distance between two points in \mathbb{R}^2 or \mathbb{R}^n). Also, we will use the notions of *metric* and *distance* interchangeably along the paper as they represent the same concept in the pattern recognition community.

The choice of the metric together with the size of the training set determine directly both *recognition rate* and *execution time*. The total execution time needed for performing a classification with T samples available in the training set and using a sampling resolution of n points will be $O(n \cdot T)$ for Euclidean and angular metrics and $O(n^2 \cdot T)$ for DTW.

As the number of training samples T increases, the recognition rate of a NN classifier should improve (as the feature space is more accurately represented). All the 3 metrics have been found to report rates above 99% with 3 or more samples loaded per gesture type [11, 23]. This is a definite advantage of NN over other classification techniques such as Rubine’s [14] for which 15+ samples are needed¹.

With respect to the other factor that determines perfor-

¹The Rubine recognizer does not employ a metric nor does it use the nearest-neighbor approach. Instead, it divides the feature space using linear boundaries that isolate gestures of the same

mance directly (the sampling rate n - see Figure 1) there seems to be little agreement between the existing works and thus little information for designers on what rate to use. Wobbrock et al. [23] propose $n = 64$ for the \$1 recognizer (which makes use of the Euclidean metric) while they also state that any value between 32 and 128 will work as fine. Li [11] reports a speed-up by using only 16 points with Protractor (the inverse cosine metric). Cao and Zhai [5] use $n = 25$ with the Euclidean metric of ShapeWriter [10] and Kratz and Rohs [8] report a value of $n = 150$ for their extension of the \$1 recognizer to 3D data. Although all these works report performance results using the same metrics, the multiple and different choices for the sampling resolution seem arbitrary and left at the intuition and experimentation of the practitioner. However, as the sampling rate influences execution time directly in a linear or quadratic manner (see Table 1), it is important to investigate its effect in practice. The goal of this work is therefore to understand the effect of sampling rate on recognizer performance in order to inform practitioners on the rates to use for pre-processing gesture input motions so that *execution time* is lowest while *recognition rate* highest. In order to achieve this goal, an analysis is conducted for the metrics commonly used with NN gesture recognizers.

1.1 Contributions

We contribute important design aspects for template-based gesture recognizers that work directly on the motion input:

1. An investigation is conducted on the sampling rate n in order to understand its effect on the recognition rate of commonly used metrics. The goal is to inform practitioners on the sampling rate to use motivated by the fact that current research report various results [5, 8, 11, 23] while n directly determines execution time;
2. We show that a linear relationship exists between the number of gestures r used by the interface and the minimum sampling rate n needed to achieve a high recognition rate. We found 6 sampling points sufficient for Euclidean and angular metrics and reasonably-sized gesture sets ($r < 30$ gestures) while a ratio of $n : r = 2 : 5$ was derived for DTW. We highlight the impact on execution time by reporting classifications that run 10 – 20 times faster than reported by existing works;
3. A theoretical argument is proposed in order to explain the experimental results and a validation test is performed on a different set of gestures. An application tool for assisting practitioners when designing their gesture recognizers is also provided as a companion for the paper.

type - therefore, it needs many samples to achieve a good space partitioning.

2. EXPERIMENT 1: THE EFFECT OF SAMPLING ON RECOGNITION RATE

Low sampling rates will make recognizers execute faster but it is likely for a trade-off to be observed with respect to recognition accuracy as fewer points are used to represent each gesture. We start by exploring the effect of sampling on recognition accuracy on the \$1 gesture set of Wobbrock et al.² [23] in order to correlate our recognition rates with those reported by previous works on the same data [11, 23]. The set is composed of 16 gesture types (Figure 2). As we progress, we will extend our analysis on more gestures.

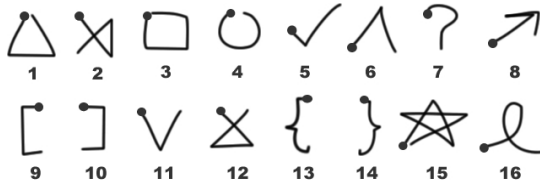


Figure 2: Gestures used for the 1st experiment [23]. 100 executions are available at normal speed for each gesture = 10 participants × 10 repetitions.

In order to test the effect of sampling over recognition rate, we consider for our analysis:

- 7 different sampling rates, $n = 4, 8, 16, 32, 64, 128, 256$. They include Wobbrock’s et al. $n = 64$ [23]; Li’s $n = 16$ [11]; but also intermediate, higher and lower resolutions;
- the 3 most commonly used metrics with NN recognizers: Euclidean [5, 10, 23], angular inverse cosine [11], and dynamic time warping [18];
- 4 progressive values for the number of training samples available for each gesture type, $T = 1, 2, 4, \text{ and } 8$ (out of the total 10 executions available for each gesture in the \$1 set).

Recognition rates were computed similarly to previous work [23] for user-dependent training scenarios: T samples were randomly chosen for training and one candidate was selected as the test for each gesture type. The process was repeated 100 times in order to compute the rate for a given metric, using a given sampling rate n , and T training samples. We report recognition rates averaged from 3 (metrics) × 7 (n) × 4 (T) × 100 (repetitions) × 16 (gestures) × 10 (participants) $\approx 1.4 \cdot 10^6$ recognition tests.

Figure 3 illustrates the mean recognition rate for each metric versus the sampling rate. In order to compare the effect on the metric alone, we only applied elementary preprocessing to the input gestures leaving out specific operations such as the golden rule search of Euclidean \$1 [23] or the optimum rotation of angular Protractor [11] which would favor one recognizer but not the others³. All gestures were therefore normalized with form factor preservation and centered to origin. As we expect from previous literature, recognition

²Available at <http://depts.washington.edu/aimgroup/proj/dollar>

³Actually, Wobbrock et al. [23] specify that their optimum alignment adds only 0.23% to recognition accuracy. Li’s Protractor [11] does the same rotation but faster. We skip these operations in order to analyze the effect on the metrics alone.

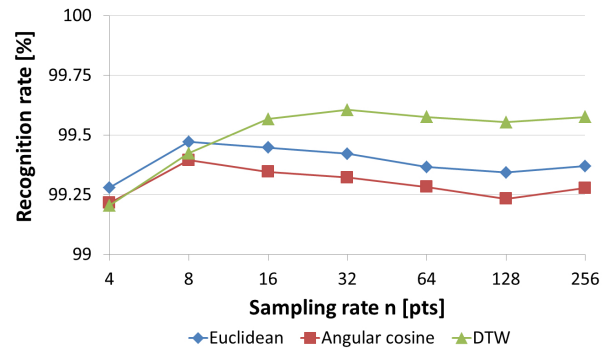


Figure 3: Recognition rate [%] versus sampling rate n [points] (the x axis shown on logarithmic scale). Note: 95% CI bars not shown on the graphic for clarity purposes but they are discussed in the text.

rates are high with a mean of 99.39%, $CI_{.95} : 99.34 - 99.45\%$. It is interesting to note that the accuracy is above 99% for all metrics and all rates including the lowest one, $n = 4$. The mean recognition rate for the lowest resolution ($n = 4$) is 99.23% which is just 0.18% smaller than the highest rate obtained (for $n = 64$) but the average gain in execution time is between 1 : 10 and 1 : 20 as showed later in the paper.

The effect of sampling rate was found significant overall, $\chi^2(6) = 46.93, p < .001$. When performing the analysis separately on each metric, the effect was significant for angular and DTW but was not significant for the Euclidean distance, $\chi^2(6) = 10.58, n.s$. This shows that, at least for the \$1 set, the Euclidean distance performing with $n = 4$ points is just as discriminative as if 32, 64, or 128 points had been used which represents a considerable reduction in execution time (given that the time complexity of the Euclidean distance is linear in n). The recognition rates did not differ significantly for $n \geq 8$ for the angular distance and $n \geq 16$ for DTW. Also, the differences between $n \in \{4, 8\}$ for angular and $n \in \{4, 8, 16\}$ for DTW were below 0.3% and showed small Cohen effects, $r < .3$.

The number of training samples T had a significant effect on recognition rate for each metric and for each sampling rate n (at $p < .001$). As expected, recognition improved from using 1 to 8 samples (with approximately 1%). No interaction effect was observed between T and n .

The results obtained on this particular set of 16 gestures show that a low sampling rate can achieve the same high recognition accuracy as much finer resolutions (specifically, $n = 4$ for Euclidean, $n = 8$ for angular, and $n = 16$ for DTW). The immediate gain is that of execution time: for linear metrics, using $n = 8$ instead of $n = 64$ would lead to a 1 : 8 reduction ratio while for DTW a reduction of 1 : 64 can be envisaged. Also, the gestures included in the \$1 set are common shapes likely to be also included by other designers in their gesture-based interfaces (such as *circle*, *rectangle*, *question-mark*, or *check*). This first experiment not only suggests that few points are feasible for the gesture metrics investigated but it also shows that such an aggressive down-sampling works very well for shapes likely to be encountered in gesture-based interfaces. However, in order not to be biased by the specific gestures of the \$1 set or its size ($r = 16$), we continue our analysis by adding more gestures and investigating multiple sets and multiple sizes.

3. EXPERIMENT 2: SAMPLING RATE VS. THE SIZE OF THE GESTURE SET

The results from the previous section were obtained on a set composed of $r = 16$ gestures. It is natural to hypothesize that a relation exists between the size of the gesture set r and the minimum sampling rate n that attains the same high recognition accuracy as much finer rates. When few gestures are present in the set (e.g. $r \leq 8$), low n rates could suffice for discrimination: even if the resolution is poor for representation, it may be enough for attaining a high recognition rate. However, when the set is large ($r \approx 30$), few sampling points may not be discriminative enough so a finer resolution should be used. Therefore, it is important to continue our investigation on which rate to use for a given size of the gesture set. For this second experiment we used additional 16 gestures acquired in similar conditions with Wobbrock et al. [23] which were selected from the set of Vatavu et al. [19] (Figure 4). Therefore, we report results from a larger set of 32 gestures. It is important to note that we are only interested in performance results for reasonably-sized gesture sets (e.g. $r \approx 30$ which represents a more than enough upper limit for the number of gestures to be included in an interface cause of human limits on learnability and memorability). These results may not apply to specific applications that need larger gesture sets such is the case of the SHARK vocabulary [10].

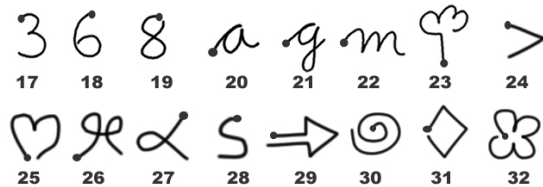


Figure 4: Additional gestures used for the 2nd experiment. Gestures are selected from [19]. Numbers continue from Figure 2.

In order to assemble data for analyzing the relation between sampling rate and set size, we used the 32 gestures as basis for generating sets with different sizes. We report results for 8 different values for $r = 4, 8, 12, 16, 20, 24, 28,$ and 30 . For each size, r gestures were chosen at random from the available 32. We considered the 3 metrics and the 4 different values for T for each set. We repeated the process 100 times for each value of r or for 100 different gesture sets or combinations out of $\frac{32!}{(32-r)! \cdot r!}$. The number of test cases was therefore $8 (r) \times 100 (r\text{-sets}) \times 3 (\text{metrics}) \times 4 (T) = 9600$ tests. For each test, we started with $n = 4$ sampling points and incremented the value of n until the recognition rate did not differ significantly from the recognition rate of $n = 64$ used as benchmark [23] (the Wilcoxon signed-rank test was used to test for significance at $p < .05$). The value of n for which the difference was not significant was stored.

Figure 5 illustrates the mean sampling rates n that achieve the same recognition accuracy as the benchmark of 64 points [23]. Linear regression models were found for each metric ($R^2 > .86$) with a slope of 0.06 for Euclidean and angular distances and 0.42 for DTW. The same intercept of 4 points was found for all metrics.

The mean rate was $n = 5.2$ points (SD=1.0) for the Eu-

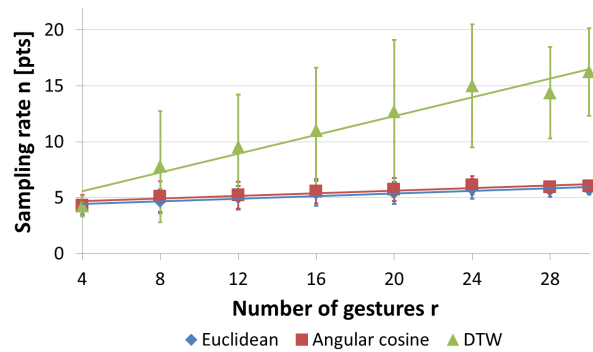


Figure 5: Sampling rate vs. the size of the gesture set. Note: error bars represent ± 1 SD.

clidean distance and $n = 5.5$ (SD=1.1) for the angular cosine. As the slope value was small (0.06), we can consider that $n = 0.06 \cdot 30 + 4 = 6$ points are sufficient for the two metrics in order to attain the same recognition rates of $n = 64$ for sets up to 30 gestures. DTW had a larger mean of $n = 11.4$ points (SD=5.1) while the regression gives $n = [0.4 \cdot r] + 4$. The stronger linear dependence showed by the DTW metric could be explained by its flexibility in the way the points of the two motions are being aligned in the search of an optimum alignment. This feature exactly makes DTW report very high recognition rates but has also made researchers find it *too elastic* for recognizing gestures. For example, Kristensson and Zhai considered the metric as a first choice for their SHARK recognizer but found in early testing that elasticity was undesirable when there were many similarly competing samples available [10]. Similar observations were noted by Wobbrock et al. [23] when reporting comparison results between DTW and $\$1$. Again, the same flexibility could explain the stronger linear dependence for the metric trying to achieve an optimal alignment.

The recognition rates and execution times for $n = 64$ and regression n values are illustrated in Figure 6 for each metric. The differences in accuracy are 0.01% for Euclidean (n.s.), 0.01% for angular (n.s.), and 0.08% for DTW while the gains in execution time are 1 : 12, 1 : 10, and 1 : 20 respectively⁴. There was a significant effect on recognition rate for DTW at $p < .001$ but the 0.08% difference is small with medium Cohen effect, $r < .5$. This can be explained by the much higher SD for DTW which makes regression n underestimate the optimum rate but by a very small difference.

In addition to these findings, we investigated whether gesture length could affect individual recognition rates. The hypothesis was that long gestures could have lower recognition rates than short ones as the n points are more scattered. We correlated individual rates with normalized lengths for all the 32 gestures. No significant Pearson correlations were found for the Euclidean metric. We found significant correlations for angular ($n \in \{4, 8, 16\}$) but their effect was too small, $R^2 < .16, p = .05$, as well as for DTW ($n \geq 16$) with also small effects, $R^2 < .27, p = .01$. Actually, the longest

⁴ $\$1$ actually takes far more time than our reported value for the Euclidean distance as it also searches for the optimum rotation for aligning two gestures. This extra operation scales the execution time by a factor of 10 ([23], p.163) which makes the overall gain to be 1:120! The gain with respect to Protractor [11], which uses $n = 16$, is smaller 6 : 16 = 1 : 2.5.

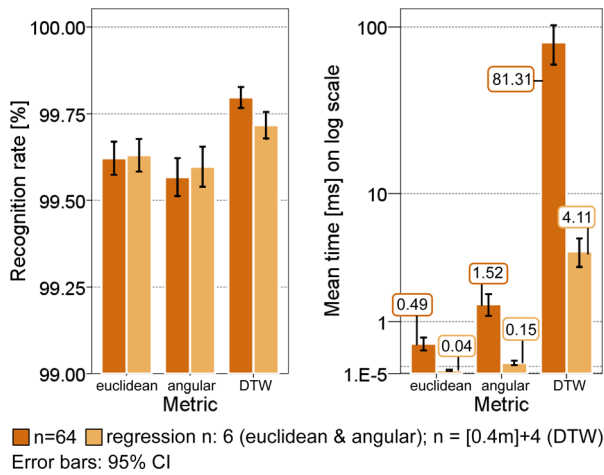


Figure 6: Recognition rates and execution times for $n = 64$ vs. regression n . Rates are averaged for all r . Execution times are computed on a 2.6 GHz P4 processor for the maximum value of $r = 32$ gestures.

gestures *15-star* and *30-spiral* attained rates of 99.9% while *14-right curly brace* (2.5 times shorter) had 97.9%. The length effects were either non significant or very small compared to the strong significant influence of the set size r .

4. A THEORETICAL ARGUMENT

The results of the reported experiments are very intriguing as they show that recognition rates above 99% can be obtained by using just a few sampled points only. Also, low sampling rates translate into low execution times for the recognizer that is able to compute the classification result more quickly.

We try to explain in the following these experimental results by proposing a theoretical argument built on the probability that two gestures are still being perceived different even when they are heavily downsampled. To this end, we introduce the notion of *probability of equivalence* for two gestures and analyze it as a function of the sampling rate. As the rate becomes lower it is more likely (more probable) that two different gestures will look similar just because the recognizer does not dispose of enough data points to discriminate on. Calculating the probability values for low sampling rates will explain the experimental results if these values are small enough for the event to occur extremely rarely in practice.

Therefore, we are set to prove that the probability of two or more gestures to be perceived similar when they are subjected to heavy downsampling (e.g. the chance of their remaining points to be located very close together) is actually very small. Although this may feel counterintuitive, we need to stress the difference between the sampling rate needed for representing a shape (for humans) and that for recognizing it (by machines). These two don't need to be (and are not) the same. While humans need more points for representation, this is not the case for the machine that matches shapes. We start our argumentation by introducing the concept of equivalence for points and curves on which we base our calculation of the probability of equivalence of gestures.

4.1 The notion of equivalence for points and curves

For all further discussion, we assume that gestures have been normalized so that they would fit in the $\mathcal{D} = [0..1] \times [0..1] \subset \mathbb{R}^2$ unit square. This is a common preprocessing step [18, 23] also being used in our two experiments.

DEFINITION Points $p, q \in \mathcal{D}$ are *equivalent* with respect to $\epsilon \in \mathbb{R}$ (and we denote $p \equiv q$) if their Euclidean distance is less than ϵ :

$$p \equiv q \Leftrightarrow \|p - q\| \leq \epsilon \quad (1)$$

Due to the fact that gestures represent continuous motions generated by the moving hand which are being artificially sampled by the acquisition device, we use in the following the formalism of plane curves.

Let \mathcal{A} be the set of all plane curves α parametrized by arc-length and taking values in \mathcal{D} :

$$\mathcal{A} = \{\alpha : [0, L(\alpha)] \rightarrow \mathcal{D} \mid \alpha(s) = (x(s), y(s)) \in \mathcal{D}\}$$

where $L(\alpha)$ is the length of α .

Let $\alpha \in \mathcal{A}$. We can sample α into n points and obtain a finite set $\alpha[n]$ by choosing n arc-lengths s_i from $[0, L(\alpha)]$:

$$\alpha[n] = \{\alpha_i = \alpha(s_i), i = 0, n-1\}$$

We refer to $\alpha[n]$ as an n -sampling of α .

Given $\alpha \in \mathcal{A}$ and $\alpha[n]$ an n -sampling of α , we can further sample $\alpha[n]$ into $m < n$ points by selecting m points out of n and obtain the subset $\alpha[m] \subset \alpha[n]$.

DEFINITION The n -samplings $\alpha[n]$ and $\beta[n]$ corresponding to curves $\alpha, \beta \in \mathcal{A}$ are *equivalent* ($\alpha[n] \equiv \beta[n]$) if all their corresponding points are equivalent:

$$\alpha[n] \equiv \beta[n] \Leftrightarrow \alpha_i \equiv \beta_i \forall i \in \{0, n-1\}$$

Figure 7 illustrates the concept of equivalence for points and gesture samplings. The constant ϵ can be chosen as fine as required. We will provide a reasonable practical estimation for ϵ later in the article.

We denote by $e(n)$ the number of pairs of equivalent points of $\alpha[n]$ and $\beta[n]$ and by $\lambda(n)$ the proportion of equivalent pairs, $\lambda(n) = e(n)/n$. Obviously, $\lambda(n) \in [0..1]$.

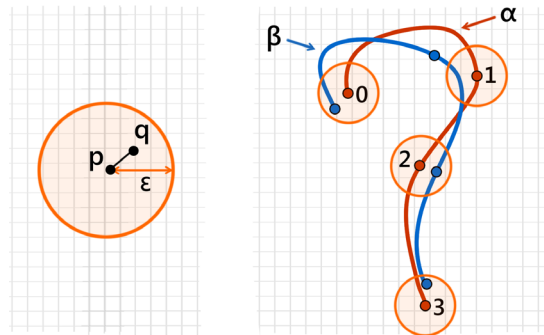


Figure 7: The concept of equivalence. Left: points p and q are equivalent with respect to ϵ if their Euclidean distance is $\leq \epsilon$. Right: gestures α and β sampled in $n = 4$ points are equivalent if all their pairs of points are equivalent.

4.2 Probability of equivalence (2 gestures)

Let $\alpha, \beta \in \mathcal{A}$ and let $\alpha[n]$ and $\beta[n]$ be their samplings into n points. The probability that $\alpha[m] \equiv \beta[m]$ for $m < n$ is given by:

$$\frac{e(n)}{n} \cdot \frac{e(n)-1}{n-1} \dots \frac{e(n)-(m-1)}{n-(m-1)} \leq (\lambda(n))^m \quad (2)$$

4.3 Probability of equivalence (r gestures)

We extend the probability of equivalence for $r > 2$ gestures by considering the probability that at least 2 gestures out of r are equivalent when sampled into m points.

Let $\lambda_{i,j}(n)$ be the proportion of equivalent pairs for gestures i and j with $i, j = 1, r, i \neq j$. The probability that the m -samplings of the two gestures are equivalent was found before as being $\leq (\lambda_{i,j}(n))^m$. Therefore, the probability for at least one pair of gestures to be equivalent has the upper bound:

$$\leq \sum_{i=1}^r \sum_{j=i+1}^r (\lambda_{i,j}(n))^m \quad (3)$$

4.4 Upper bounds for the probability of equivalence

We are interested in estimating the values of $\lambda(n)$ in order to calculate upper bounds for equations 2 and 3.

THEOREM The probability that two points $p, q \in \mathcal{D}$ are equivalent can be approximated as follows:

$$P(p \equiv q) \approx \pi \epsilon^2 \quad (4)$$

Proof. The probability function of the Euclidean distance between two points picked at random in the unit square is $P(t) = 2t(t^2 - 4t + \pi)$ [21]. Therefore, the probability that the distance between p and q is at most ϵ will be:

$$P(t \leq \epsilon) = \int_0^\epsilon P(t) dt = \frac{1}{2} \epsilon^4 - \frac{8}{3} \epsilon^3 + \pi \epsilon^2$$

which can be approximated by $\pi \epsilon^2$ for small $\epsilon \leq 1$. \square

If the n pairs of points are chosen independently from α and β then the random variable $e(n)$ follows the Binomial distribution with the probability of success $p = \pi \epsilon^2$. Therefore, the expected value for $e(n)$ is $E[e(n)] = np$ [7](p.181) which makes the expected value for $\lambda(n)$ to be p . However, we are interested in the expected value of $(\lambda(n))^m$ that appears in equations 2 and 3:

$$\begin{aligned} E[(\lambda(n))^m] &= E\left[\left(\frac{e(n)}{n}\right)^m\right] \\ &= \frac{1}{n^m} E[e^m(n)] \\ &= \frac{1}{n^m} \sum_{k=0}^n k^m \binom{n}{k} p^k (1-p)^{n-k} \quad (5) \end{aligned}$$

THEOREM The sum $S(n, m)$ of equation 5 is a polynomial in n of order m with the coefficient of n^m being p^m .

Proof. By using $\binom{n}{k} = \frac{n}{k} \binom{n-1}{k-1}$, putting $k-1 = t$ and expanding $(1+t)^{m-1}$ in (5), it can be easily showed that:

$$S(n, m) = np \sum_{i=0}^{m-1} \binom{m-1}{i} \cdot S(n-1, i)$$

with $S(n, 0) = S(n-1, 0) = 1$.

Next, the principle of induction can be used. For $m = 1$, $S(n, 1) = np$ and for $m = 2$, $S(n, 2) = n^2 p^2 + n(p-p^2)$. We assume that $S(n, m) = n^m p^m + O(n^{m-1})$ where by $O(n^{m-1})$ we denote a polynomial in n of order $m-1$.

$$S(n, m+1) = np \sum_{i=0}^m \binom{m}{i} \cdot (p^i (n-1)^i + O(n^{i-1}))$$

in which n^m appears once for $i = m$ with the coefficient $\binom{m}{m} p^m$. Therefore, $S(n, m+1) = n^{m+1} p^{m+1} + O(n^m)$. \square

Using these findings, we can write:

$$E\left[\left(\frac{e(n)}{n}\right)^m\right] = \frac{S(n, m)}{n^m} = \frac{p^m n^m + O(n^{m-1})}{n^m}$$

If we consider n to be large enough⁵ then we can approximate the expected value of $(\lambda(n))^m$ to p^m . We can further use Markov's inequality [7](p.437) in order to obtain an upper bound value λ for $(\lambda(n))^m$ for which the probability $P((\lambda(n))^m \geq \lambda)$ can be neglected (e.g. the probability is less than ≤ 0.01):

$$P((\lambda(n))^m \geq \lambda) \leq \frac{E[(\lambda(n))^m]}{\lambda} \approx \frac{p^m}{\lambda} = 0.01$$

Therefore, by choosing $\lambda = 100 \cdot p^m$, we are 99% confident that $(\lambda(n))^m \leq \lambda$. By using this upper bound in equation 3, we obtain an upper bound for the probability of equivalence of the m -samplings of r gestures:

$$P(m, r) \leq 50 \cdot r(r-1) \cdot p^m = 50 \cdot r(r-1) \cdot (\pi \epsilon^2)^m \quad (6)$$

The only unknown factor so far is the constant ϵ that actually defines the equivalence of two points (see equation 1) for which we still need to choose a suitable value. A simple method to estimate ϵ is to look at the data sets from experiments 1 and 2 and average the Euclidean distances for all pairs of points that *should be equivalent*, i.e. for all the points that occupy the same index locations on gestures of the same class. After performing the computation, an average value of $\epsilon = .08$ (SD=.02) was obtained. We can therefore safely choose an upper bound value of $\epsilon = .1$ for all our further calculations. Equation 6 becomes:

$$P(m, r) \leq 50 \cdot r(r-1) \cdot 0.0314^m \quad (7)$$

In practice, template-based recognizers work with several samples T stored for each gesture class. Therefore, we need to consider T for each of the r gesture classes which makes the upper bound of equation 7 become:

$$P(m, r) \leq 50 \cdot r \cdot T \cdot (r-1) \cdot T \cdot 0.0314^m \quad (8)$$

Figure 8 illustrates the values for the probability of equivalence $P(m, r)$ for different sampling rates m and set sizes r . Values were computed for $T = 8$ which is a more than reasonable value (Wobbrock et al. [23] showed that recognition rates above 99% can be obtained for $T \geq 3$ samples per class). For $m = 5$, the upper bound of the probability is $\approx 8\%$ for sets of $r = 30$ gestures, drops down to 2% for $r = 20$, and stays under 1% for smaller sizes. For $m \geq 6$, the probability stays under 0.3% for all $r \leq 30$ and becomes $\leq 0.01\%$ for $m \geq 7$. These values support our experimental results by showing that the probability for any two gestures

⁵we can safely make this assumption as n can be viewed as a very fine sampling of a continuous curve from \mathcal{A} .

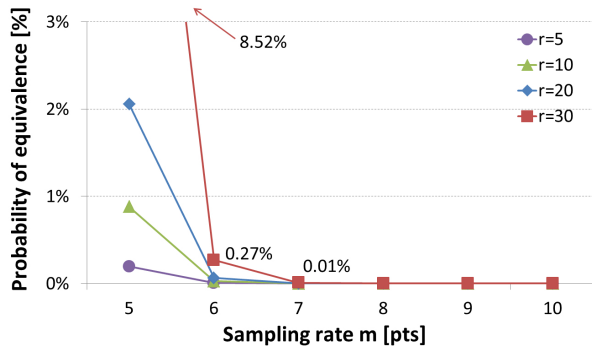


Figure 8: Upper bounds for the probability of equivalence for sets of size r , sampling rates m , and $T = 8$ training samples (see equation 8).

to be perceived similar cause of downsampling is very small. Although there might exist differences with respect to these theoretical values caused by constraints related to gesture production (e.g. we made the assumption that points are drawn randomly from the unit square), our theoretical calculations provide at least some evidence that the probability of equivalence is very small even for low sampling rates.

5. EXPERIMENT 3: VALIDATION

In order to validate our results, we performed a new recognition experiment using the UJIPenchars2 pen strokes set [12] available from the UCI Machine Learning Repository⁶ [6]. The set consists in 97 gestures including 10 digits, 52 English letters (lower and upper case), 14 Spanish characters and other 21 strokes including various punctuation marks. The total number of available samples is $11,640 = 60$ participants \times 97 gestures \times 2 repetitions.

The purpose of this experiment was to validate the sampling rates suggested by the regression models of Figure 5 for other gestures. We used the same experimental plan as before with 4 sizes $r = 10, 20, 30, 40$; 100 repetitions for each r (for each repetition r gestures were randomly chosen out of the 97 available); 1 training and 1 testing sample (limited by the set); 2 repetitions (one for each training sample). Therefore, we had 3 (metrics) \times 4 (r) \times 100 (r -sets) \times $2 = 2400$ tests. The sampling rate n was automatically computed from the regression models (to which we refer to as model n) and the recognition rate was tested for significance against that of the benchmark of 64 points.

Results showed a very good fit of the linear model for the Euclidean and angular metrics. For the Euclidean, the recognition rate of model n was not significantly different from $n = 64$ in 98% of the tests (at $p < .05$). Similarly, no significant difference was observed for the angular metric in 99% of the tests. The average rates were 6 points (SD=.71) for Euclidean and 6.5 points (SD=.50) for angular cosine.

For the DTW however, only 68% of the tests showed n.s. differences between the recognition rates. As discussed in experiment 2, the larger SD for DTW could account for this result. The difference however was of just 0.5%. Nevertheless, another model could better fit the behaviour of the DTW metric in the sense that larger values should be suggested for the sampling resolution n . For this purpose, we analyzed

⁶<http://archive.ics.uci.edu/ml>

the frequency distribution of n for DTW using data from experiment 2 and, instead of computing the mean rate for each size r (as in Figure 5), we computed an upper threshold for n so that the probability of n falling beyond the threshold becomes non significant (.05) which we denoted $n_{.95}$. In other words, there are 95% chances that n will fall $\leq n_{.95}$. Using these rates we derived a more tight model for DTW for which a logarithmic regression showed the strongest fit ($R^2 = .86$):

$$n_{.95} = 8.47 \cdot \ln(r) - 3.79 \quad (9)$$

When using the $n_{.95}$ threshold for the cases in which a significant difference was noticed between the recognition rates of DTW, the total percentage of n.s. tests increased to 96%. We also tried the $n_{.95}$ approach for the Euclidean and angular metrics as well but we found a non-interesting value of 7 points in both cases (vs. 6 obtained with the linear model). Interestingly, the experiment showed that the linear relationship works and extrapolates very well for these two metrics (note that a set of size $r = 40$ was also used).

6. CONCLUSIONS

An investigation was conducted on the effect of motion sampling rate on the performance of recognizers that HCI practitioners will likely use in order to develop gesture-based interfaces. We found that 6 points are sufficient for attaining high recognition rates for Euclidean and angular distances while both a linear and logarithmic relationship with the size of the gesture set were derived for DTW. Our findings relate to the *peaking phenomenon* encountered in pattern recognition [17] that describes the situation in which adding more features up from one point does not improve but actually increases the classification error.

We must stress that this work does not address the recognition of very large gesture sets (e.g. the 10k set of [10] for which special techniques were developed). Instead, we focus on the practical needs of designers that would likely choose between $\$1$ or DTW as recognizers and design reasonably sized sets of 10-20 gestures for their interfaces. Although our observations were derived for sets up to 30 gestures, this upper limit is usually enough for common gesture-based interfaces (as limited by the number of gesture commands users can learn and recall).

6.1 A note on extreme under-sampling

The recognition rates above 99% obtained using just few sampling points are very intriguing. As a more intriguing note, we also investigated $n = 2$ (first and last points of each gesture) and $n = 3$ (first, middle, and last) which simplify gestures at the most extreme levels. Recognition rates of 87.8% and 95.5% were obtained on the set of 32 gestures from Figures 2 and 4 using $n = 2$ and $n = 3$ points respectively. This confirms the preprocessing operation of Kristensson and Zhai [10] (p.45) that pruned their massive 10k database of strokes using a filter on the first and last points of each gesture (which equivalates to using the Euclidean distance with $n = 2$ sampling points) but they do not give numeric data on how good this pruning actually is. However, there is distinction between sampling for recognition (which is the goal of this work) and sampling for filtering [10]. Although such extremely aggressive downsampling showed high recognition rates for our gestures, it appears too extreme to be used for other purposes than filtering.

6.2 Sampling Rate Analysis Tool

Our analysis showed that the sampling rate can significantly reduce the processing time of gesture recognizers. Small rates of 6 points were found sufficient for attaining high recognition rates. However, in practice and with different gestures, 6 points may not always prove the best minimum choice. Therefore, the appropriate sampling rate for a given gesture set and metric will need to be determined by the practitioner. In order to assist this process, a simple tool was developed for helping practitioners working out the sampling resolution that would attain the highest recognition rate as well as the lowest execution time. This follows the tradition of providing assisting tools for designers of gesture-based interfaces in the line of the gesture development toolkit *gdt* of Long et al. [13] or the MAGIC application of Ashbrook and Starner [4]. The application includes the regression models derived in the paper but also computes recognition rates and execution times for other sampling rates. At the end of the analysis, it will provide the C# class representing the already implemented NN recognizer with the provided metric and sampling rate. The application can be downloaded from the author's web page⁷.

6.3 Future work

Future work could investigate the effect of non-uniform vs. uniform samplings (such as samplings resulted from polyline simplification algorithms or from identifying corners or high curvature points on the motion input). Also, it remains to investigate whether the same small rates still apply for 3D gestures.

It is important to consider this work as the first exploration of sampling rate and its effects on recognizer performance. We hope that the results will prove useful to practitioners concerned with performance issues when implementing gesture-based recognizers.

7. ACKNOWLEDGMENTS

This paper was supported by the project "Progress and development through post-doctoral research and innovation in engineering and applied sciences- PRiDE - Contract no. POSDRU/89/1.5/S/57083", project co-funded from European Social Fund through Sectorial Operational Program Human Resources 2007-2013.

The author would also like to thank mathematician Bianca Satco for verifying a first draft of the theoretical argument described in Section 4.

8. REFERENCES

- [1] D. Anderson, D. Bailey, and M. Skubic. Hidden markov model symbol recognition for sketch-based interfaces. In *AAAI Fall Symp.: Making Pen-Based Interaction Intelligent and Natural*, pages 15–21, 2004.
- [2] L. Anthony and J. O. Wobbrock. A lightweight multistroke recognizer for user interface prototypes. In *GI '10*, pages 245–252, 2010.
- [3] C. Appert and S. Zhai. Using strokes as command shortcuts: cognitive benefits and toolkit support. In *CHI '09*, pages 2289–2298, 2009.
- [4] D. Ashbrook and T. Starner. Magic: a motion gesture design tool. In *CHI '10*, pages 2159–2168, New York, NY, USA, 2010. ACM.
- [5] X. Cao and S. Zhai. Modeling human performance of pen stroke gestures. In *CHI '07*, pages 1495–1504, 2007.
- [6] A. Frank and A. Asuncion. UCI machine learning repository, 2010.
- [7] S. Ghahramani. *Fundamentals of Probability, 2nd Ed.* Prentice Hall, 2000.
- [8] S. Kratz and M. Rohs. A \$3 gesture recognizer: simple gesture recognition for devices equipped with 3d acceleration sensors. In *IUI '10*, pages 341–344, 2010.
- [9] S. Kratz and M. Rohs. Protractor3d: a closed-form solution to rotation-invariant 3d gestures. In *IUI'11*, pages 371–374, New York, NY, USA, 2011. ACM.
- [10] P.-O. Kristensson and S. Zhai. Shark2: a large vocabulary shorthand writing system for pen-based computers. In *UIST '04*, pages 43–52, 2004.
- [11] Y. Li. Protractor: a fast and accurate gesture recognizer. In *CHI '10*, pages 2169–2172, 2010.
- [12] D. Llorens and et al. The UJpenchars database: A pen-based database of isolated handwritten characters. In *Proc. 6th Int. Conf. on Language Resources and Evaluation*, 2008.
- [13] A. C. Long, Jr., J. A. Landay, and L. A. Rowe. Implications for a gesture design tool. In *CHI '99*, pages 40–47, New York, NY, USA, 1999. ACM.
- [14] D. Rubine. Specifying gestures by example. In *SIGGRAPH '91*, pages 329–337, 1991.
- [15] T. Sebastian, P. Klein, and B. Kimia. On aligning curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(1):116–125, 2003.
- [16] T. M. Sezgin and R. Davis. Hmm-based efficient sketch recognition. In *IUI '05*, pages 281–283, New York, NY, USA, 2005. ACM.
- [17] C. Sima and E. Dougherty. The peaking phenomenon in the presence of feature-selection. *Pattern Recognition Letters*, 29:1667–1674, 2008.
- [18] R.-D. Vatavu, L. Grisoni, and S.-G. Pentiu. Gesture recognition based on elastic deformation energies. In *LNCS 5085*, pages 1–12. Springer Berlin / Heidelberg, 2009.
- [19] R.-D. Vatavu, D. Vogel, G. Casiez, and L. Grisoni. Estimating the perceived difficulty of pen gestures. In *INTERACT'2011, Part II, LNCS 6947*, pages 89–106. Springer, 2011.
- [20] A. Webb. *Statistical Pattern Recognition*. John Wiley & Sons, Inc., 2002.
- [21] E. W. Weisstein. Square line picking. from mathworld—a wolfram web resource. <http://mathworld.wolfram.com/squarelinepicking.html>.
- [22] D. Willems, R. Niels, M. v. Gerven, and L. Vuurpijl. Iconic and multi-stroke gesture recognition. *Pattern Recognition*, 42(12):3303–3312.
- [23] J. O. Wobbrock, A. D. Wilson, and Y. Li. Gestures without libraries, toolkits or training: a \$1 recognizer for user interface prototypes. In *UIST '07*, pages 159–168, 2007.

⁷<http://www.eed.usv.ro/~vatavu>