PAPER

Online Continuous Scale Estimation of Hand Gestures

Woosuk KIM[†a)], *Nonmember*, Hideaki KUZUOKA[†b)], *and* Kenji SUZUKI[†,††c)], *Members*

SUMMARY The style of a gesture provides significant information for communication, and thus understanding the style is of great importance in improving gestural interfaces using hand gestures. We present a novel method to estimate temporal and spatial scale—which are considered principal elements of the style—of hand gestures. *Gesture synchronization* is proposed for matching progression between spatio-temporally varying gestures, and scales are estimated based on the progression matching. For comparing gestures of various sizes and speeds, gesture representation is defined by adopting *turning angle representation*. Also, LCSS is used as a similarity measure for reliability and robustness to noise and outliers. Performance of our algorithm is evaluated with synthesized data to show the accuracy and robustness to noise and experiments are carried out using recorded hand gestures to analyze applicability under real-world situations.

key words: *hand gestures, gesture synchronization, scale estimation, longest common subsequence (LCSS)*

1. Introduction

Using hand gestures to improve human-computer interfaces has been of great interest for an expressive and natural way of interaction [1]. As human behavior, the meaning of expression made by hand gestures can be interpreted in two main ways—*what* is done and *how* it is performed [2]. Traditional gestural interfaces have usually been for recognizing the former (a message or content).

However, style—the way in which a gesture is performed—also has significant importance in communication. For instance, stylistic differences can be used to describe characteristics of individual personality [2], and changes in the style may imply transition in emotional status [3], [4]. Our interest is in expanding perspectives of gesture recognition by taking into account style of gestures and improving expressiveness of interfaces for applications like 3D animation. Especially, we focus on the dynamics of hand gestures measured in temporal and spatial scale variation, which is an essential element contributing to stylistic differences [2].

We have chosen single-handed motions as targets for estimating scale. Single-handed motions can be thought of as spatio-temporal trajectories or shapes in space. So it is less ambiguous to define scale measures on them, and various methods for representation and comparison of trajectories and shapes are introduced by many researchers [5]. Even though single-handed motions may not be complex gestures, estimating their scale is not a simple task. Comparing time series or temporal sequence with variations in size and speed requires, at the very least, representation methods which preserve scale-related information and similarity measures independent from scale variations.

In addition, our method assumes real-time evaluation of scales from online continuous user input streams, which can improve interactivity by minimizing delay between interactions and providing localized scale information since many applications using gestural interfaces are required to behave in a highly interactive manner (like multimedia and entertainment applications). This requirement adds difficulties to the problem since boundaries (start and end points of gestures) are unknown for data from continuous input streams [6]. *Gesture spotting* [7]–[10] in continuous gesture recognition presents similar difficulties but with different requirements; we are interested in on-going local progression of gesture input for continuous scale evaluation, however *gesture spotting* tries to isolate single and whole gesture data for consecutive recognition without explicit intervention by users or systems.

2. Related Work

Analysis on auxiliary information of gestures

Extracting and using auxiliary information of gestures has been great a concern of researchers in various fields. Wilson and Bobick [11] put emphasis on systematic understanding of variations in gestures, as well as recognition. They introduced *parametric hidden markov models* (PHMMs), which embed spatial parameters in *hidden markove models* (HMMs). The spatial variation of gestures may include size, direction, and so forth. Herzog et al. [12] extended this method to analyze human arm motions and synthesize the motions using robot arms. However, PHMMs are used to parameterize only spatial variations but not temporal variations. Appert and Bau [13] specifically focused on scale detection in early stages of the recognition process. To measure scale difference explicitly, they proposed a gesture representation method inspired by a *turning angle representation* in shape matching. Since the representation holds scale

Manuscript received December 19, 2011.
Manuscript revised May 9, 2012.

[†]The authors are with the Faculty of Engineering, Information and Systems Division of Intelligent Interaction Technologies, University of Tsukuba, Tsukuba-shi, 305–8577 Japan.

[††]The author is with the Japan Science and Technology Agency (JST), Kawaguchi-shi, 332–0012 Japan.

a) E-mail: s0730443@u.tsukuba.ac.jp
b) E-mail: kuzuoka@iit.tsukuba.ac.jp
c) E-mail: kenji@ieee.org

DOI: 10.1587/transinf.E95.D.2447

information of gestures measured as distance, it is not suitable for estimating temporal variations of gestures.

Various aspects of gestures are perceived by not only low level features like size and speed, but also high level features described by more abstract terms. Camurri et al. [14], [15] attempted to analyze expressiveness of human motions or gestures using a layered approach. From low level features gathered from multiple cues, the system outputs emotional status like anger, fear and joy. We share similar ideas in interpreting gestures (style and expressiveness can be interchangeable in many contexts). However their analysis is of overall behaviors rather than specific gesture types. Rehm et al. [16] introduced another interesting work, which differentiate cultural influence from gestures.

Using human motions to improve naturalness and expressiveness of synthesized characters has a long history in 3D animation research. Some of the approaches use variations on gestures for changing style of animation. Thorne et al. [17] adopted sketching gestures to control character motions. They defined gesture vocabulary, which is used to map between gestures and motions. Different from typical performance animation, their system identifies gesture input to select motion types and adjusts motions with the variations of gestures like height and width of symbols or timing of drawing. Shiratori and Hodgins [18] used accelerometer-based motion sensing devices to control a physically simulated character. Changes in frequency, amplitude, and so on are used to variate locomotion of a character. Both are distinctive to traditional performance animation, since they recognize what users performed and variations of gestures.

Similarity measures

A good review of similarity measures for shapes is provided by Veltkamp [5]. One of these methods, *Turning angle distance*, compares shapes independently from translation, rotation and scaling [13], [19], [20]. As a similarity measure for temporally mismatching data, *dynamic time warping* (DTW) has been widely adopted by various applications like speech/gesture recognition, image matching and so forth [21]–[23]. Vlachos et al. [24] introduced a spatio-temporally invariant similarity measure for 2D trajectories by combining *turning angle distance* and DTW. Recently, LCSS has been gathering interest as an alternative to DTW for its robustness to noise [25], [26]. Although LCSS is originally for string matching, several modifications have been suggested for trajectory matching [27]–[29].

Evaluating gesture progression

Bevilacqua et al. [30] proposed a method, referred to as *gesture following*, to evaluate temporal progression of gestures using HMMs. Their approach is analogous to incremental search with windowing in gesture spotting algorithms [7]–[10] but its focus is shifted from a segmenting task [6] of continuous gesture to tracing local progression of on-going gestures. Mori et al. [31] introduced similar incremental search algorithm but with DTW to predict subsequent gesture motion. However, in both [30] and [31], variation of spatial and temporal scale is not taken into consideration.

3. Gesture Representation and Scale Measure

3.1 Gesture Representation

Trajectory data which is tracked from single-handed motions may differ in its location, scale and rotation, even for gestures of the same type. These inconsistencies are usually caused by: 1) physical characteristics of subjects like height, arm length and so forth, 2) changes in expression, 3) sensor configuration—position and orientation. Since we are focusing on estimating scale of gestures, scale and translation independent gesture representation are necessary (for rotation, we assume similar gesture shapes with different rotation and orientation are not the same). Furthermore, scale related information has to be preserved in the representation to explicitly calculate scale differences.

For scale and translation invariant representation of gestures, we have adopted ideas of *turning angle representation* [13], [19], [20]. In *turning angle representation*, a shape or trajectory is defined as a cumulative angle function or *turning function* between consecutive polylines. Comparing turning angles between shapes with same perimeter (after normalization) makes it scale independent. Similarly, our approach takes angular changes into account for gesture description. Where a trajectory of single-handed gesture motion G in 3D space is described as a temporal sequence of position vectors $\overrightarrow{p} = (x, y, z)$ (Eq. (1)), a displacement vector $\overrightarrow{d}$ defines a change in both angle and distance from ith frame to $(i+1)$th frame. Then, the gesture G can be re-written as a sequence of displacement vectors $\overrightarrow{d}_i = \overrightarrow{p}_{i+1} - \overrightarrow{p}_i$ to describe consecutive angular changes for the whole gesture sequence (Eq. (2)) like in Fig. 1.

$$G = \overrightarrow{p}_1, \overrightarrow{p}_2, \ldots, \overrightarrow{p}_t, \ldots, \overrightarrow{p}_N \quad (1)$$

$$= \overrightarrow{d}_1, \overrightarrow{d}_2, \ldots, \overrightarrow{d}_i, \ldots, \overrightarrow{d}_{N-1} \quad (2)$$

This representation is translation invariant since displacement vectors are calculated relatively between adjacent frames (independent from its starting position) and invariant in spatial scale by only comparing angles between displacement vectors, and spatial information is preserved as length of displacement vectors also. In addition, applying elastic

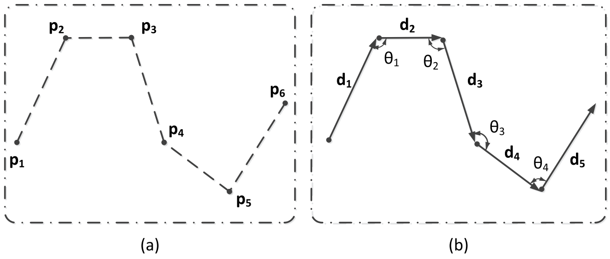


Fig. 1 An example of gesture representation: A sequence of points (a) is converted into a sequence of displacement vectors (b) and a sequence of θ is considered changes in turning angles of a gesture trajectory.

matching like DTW or LCSS makes it possible to be temporal scale independent, which is explained in more detail in Sect. 4.1.

3.2 Scale Measure

It is difficult to define absolute measure for scale differences, since style is subject to various factors like personality, culture and so forth [2], [16]. So in our approach, scale variation is measured in a relative way. That is, scale of user input is compared to pre-defined or recorded reference gestures.

For gesture sequences represented as Eq. (2), basically, temporal length is measured in number of elements $(N-1)$ and spatial length is measured in arc length $(\sum \left|\overrightarrow{d}_i\right|)$. The ratio of these lengths between a reference gesture and user input describes scale variations.

In addition, the following ideas are taken into consideration in our algorithm:

- global and local measurement: Local changes in scale of gestures also affect the style. For instance, performing a gesture with incremental speed may express acceleration. It is required to measure scales on limited interval for estimating localized scale variations.
- excluding outliers: Gesture data may contain outliers, which are caused by either sensor noise or human error like hand shaking. For more accurate scale estimation, it is desirable to exclude such outliers.

4. Algorithm

The estimation process mainly consists of the following three steps:

1. preparing a reference gesture: Single-handed motion is recorded and smoothed (e.g. by Gaussian or kalman filters) to reduce noise. Then the recorded sequence of position vectors is converted into a sequence of displacement vectors (Eq. (2)). Scale of the reference gesture is considered as unit scale.
2. gesture synchronization: After a user starts gesture performance, amounts of progression by accumulated (and possibly incomplete) input data are evaluated. The evaluation of gesture progression is carried out by searching for corresponding segments in the reference and user input, which are spatio-temporally matching each other.
3. estimating scale: Temporal and spatial scale of user input are estimated using the progression matching given by step 2. Scale of the user input is expressed in ratio to scale of the reference gesture. Step 2 and 3 are repeated until gesture performance is finished.

We assume the type of gesture is known before starting gesture performance since this paper focuses on estimation rather than recognition. Approaches for simultaneously applying gesture recognition and scale estimation will be discussed in Sect. 6.

4.1 Similarity Measure

In this paper, as an alternative to the DTW-based approach in our previous work [32], LCSS is used to define a similarity measure. In general, LCSS is known to be more robust to noise than DTW since outliers, possibly caused by noise, can be ignored [25], [26]. On the contrary, in DTW, by its definition, every element must have one or more matching elements whether they are similar or not. This means warp-paths built by DTW may contain ill-matched pairs of elements under noisy conditions, which results in inaccurate distance calculation.

Our similarity measure is motivated by other LCSS-based methods for motion sequence and trajectory matching [27]–[29] but also with modification under our own requirements, as below:

- As mentioned in Sect. 3.1, a gesture is represented as a sequence of displacement vectors and only their angular changes are taken into consideration for scale invariant comparison. For comparing angular similarity, we use a cosine distance and a matching function with a threshold. If the distance is smaller than the threshold, they are said to be matching. For two displacement vectors $\overrightarrow{A}$ and $\overrightarrow{B}$, the distance function d and matching function *match* are described as:

$$d(\overrightarrow{A}, \overrightarrow{B}) = 1 - \frac{\overrightarrow{A} \cdot \overrightarrow{B}}{\left|\overrightarrow{A}\right| \left|\overrightarrow{B}\right|} \tag{3}$$

$$match(\overrightarrow{A}, \overrightarrow{B}) = \begin{cases} true, & \text{if } d(\overrightarrow{A}, \overrightarrow{B}) < d_{threshold} \\ false, & otherwise \end{cases} \tag{4}$$

- Two elements of a matching pair are not necessarily identical because matching is decided on cosine distance as above. Hence, it is necessary to express the longest common subsequence as a sequence of matching pairs like a warp-path in DTW. For given sequences $X_i = x_1, x_2, \ldots, x_i$ and $Y_j = y_1, y_2, \ldots, y_j$, the modified LCSS with warp-path $W_k = \mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_k$, where $\mathbf{w}_k = (w_k^X, w_k^Y)$ is defined as:

$$LCSS(X_i, Y_j) = \begin{cases} \emptyset, \quad \text{if } i=0 \; or j=0 \\ (LCSS(X_{i-1}, Y_{j-1}), (x_i, y_j)) \text{ and } W_k = (W_{k-1}, (i, j)), \\ \qquad \text{if } match(x_i, y_j) \text{ is } true \\ longest(LCSS(X_i, Y_{j-1}), LCSS(X_{i-1}, Y_j)) \\ \qquad otherwise \end{cases} \tag{5}$$

4.2 Gesture Synchronization

To estimate the scale of an incomplete gesture from continuous data streams, a segment in a given reference gesture sequence, which corresponds to partial user input, must be found prior to comparison (Fig. 2). In other words, it is necessary to evaluate on-going spatio-temporal progression of

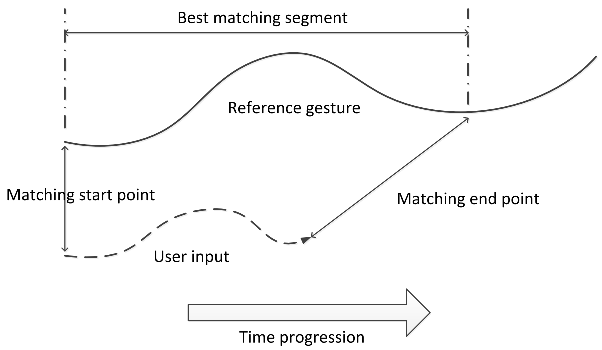


Fig. 2 For a given reference gesture and user input, *gesture synchronization* finds start and end point of the reference gesture, which describe the most similar segment to the user input.

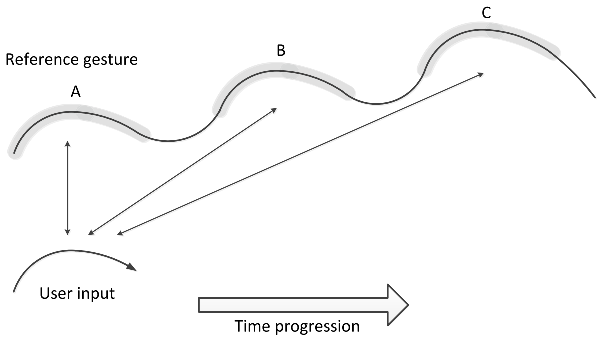


Fig. 3 If a reference gesture has repeated segments A, B and C, user input can be matched to all or any of them (depending on implementation) in LCSS. However, it is natural to think of segment A as the best matching since user input is on-going and there may be more input like B and C along with progression.

user input in relation to the reference gesture; we refer to this process as *gesture synchronization*.

LCSS makes this process quite simple since its results are longest (or with maximum similarity) common sequences by themselves. For two sequences X and Y, the first element of the longest common sequence $C = LCSS(X, Y)$ will be a start point, and the last element represents an end point of the matching boundary.

The overall process of *gesture synchronization* is described as follows:

1. For a given reference gesture R and partial user input U, subsequence R_{win} and U_{win} are defined by a sliding window algorithm (Sect. 4.2.1).
2. LCSS is performed on the two sliding windows. The resulting sequence $W = LCSS(R_{win}, U_{win})$ represents spatio-temporal matching between the sliding windows and the last matching pair of W is appended to a sequence W_{global}, which describes gesture progression for the whole user input.
3. W_{global} is expanded for compensating ignored matching caused by LCSS (Sect. 4.2.2).

4.2.1 Sliding Window

We apply a sliding window algorithm for both reference gestures and user input in *gesture synchronization* process with the following reasons:

- computational efficiency: Time complexity of LCSS, when comparing two sequences of n and m elements, is $O(n \cdot m)$ in naive approaches. That is, the amounts of time required for comparing gesture sequences increase along with growing size of user input. So, it may not be guaranteed to evaluate gesture progression in constant time (e.g. sampling interval), which is not desirable for interactive applications. Also, the similarity measure for elements is quite a costly operation since it compares 3D vectors using cosine distance. This may be critical for real-time interaction in certain hardware platforms, especially mobile devices.
- preventing ill-matching: When lengths of two temporal sequences are greatly differ, LCSS may produce unexpected results. For instance, if a reference gesture has repetitive patterns, partial user input which is similar to the patterns can be matched to any of them (Fig. 3). It is reasonable to think that matching should occur in temporal order however LCSS can not differentiate them. It is possible to prevent this kind of ill-matching by limiting size of sequences to be compared.

Size of sliding windows should be small enough for computational efficiency, but also large enough for reliability of evaluation. The optimum size may differ by application requirements and hardware characteristics (like cpu power, sampling rates, sensor noise and so forth) thus it has to be decided by empirical methods.

Also, as we assumed spatio-temporal deformation of gesture sequences, both sliding windows of the reference and user input are set to include the same amounts of gesture progression data not the same number of elements. The maximum ratio between the size of two sliding windows varies depending on the range of scale variation required by applications.

For a given reference gesture

$$R = \overrightarrow{r}_1, \overrightarrow{r}_2, \ldots, \overrightarrow{r}_N\ , \tag{6}$$

user input at time t

$$U = \overrightarrow{u}_1, \overrightarrow{u}_2, \ldots, \overrightarrow{u}_t \tag{7}$$

and a warp-path at time $(t-1)$

$$W_k = \mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_k,\ \text{where}\ \mathbf{w}_k = (w_k^R, w_k^U)\ , \tag{8}$$

let M_{win} and M_{ratio} be a maximum window size and a maximum size ratio between sliding windows respectively then start and end index of the user input window (s_U, e_U) are defined as:

$$(s_U, e_U) = \begin{cases} (1, t), & k \leq M_{win} \\ (w^U_{k-M_{win}}, t), & otherwise \end{cases} \tag{9}$$

and start and end index of the reference window (s_R, e_R) are

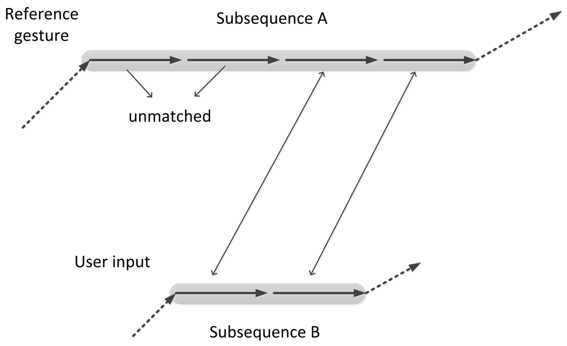


Fig. 4 Even if subsequence B is considered to be matched to the whole subsequence A, only two elements of the subsequence B become parts of matching in LCSS.

defined as:

$$(s_R, e_R) = \begin{cases} (1, \lceil t \times M_{ratio} \rceil), & k \leq M_{win} \\ (w^R_{k-M_{win}}, min(s_R + \lceil (e_U - s_U) \times M_{ratio} \rceil, N)), & otherwise \end{cases} \tag{10}$$

4.2.2 Correcting Warp-Path

For sampled sequences of continuous trajectories, one element may have several matching elements because of spatio-temporal deformation. However, in LCSS, one element can have only one matching element as mentioned in Sect. 4.1. As a consequence, elements which are considered parts of matching may be ignored as outliers (Fig. 4). For example, if a reference sequence is $R = (a, a, a, b, b, c, c)$ and a user input sequence is $U = (a, a, b, b, b, c, c)$, it is reasonable to think the subsequence $R_3 = (a, a, a)$ is matching to the subsequence $U_2 = (a, a)$ since we consider a gesture as a continuous trajectory rather than a sequence of discrete symbols (R_3 and U_2 hold same amount of gesture progression data but are spatio-temporally stretched). To preserve scale information properly, it is necessary to convert a warp-path described as a sequence of matching elements into a sequence of matching subsequences by merging adjacent similar elements into a subsequence. A warp-path W (Eq. (8)) is converted into a corrected warp-path P (Eq. (11)) by a procedure shown in Algorithm 1.

$$\begin{aligned} P = \mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_k, & \\ \text{where } \mathbf{p}_k = (p^R_k, p^U_k) & \\ = ((w^R_k - a_k, w^R_k - a_k + 1, \ldots, w^R_k + b_k - 1, w^R_k + b_k), & \\ (w^U_k - c_k, w^U_k - c_k + 1, \ldots, w^U_k + d_k - 1, w^U_k + d_k)) & \end{aligned} \tag{11}$$

4.3 Estimating Scales

Temporal and spatial scale of input gestures are estimated based on the gesture progression expressed in expanded warp-path P (Eq. (8)). Let $seqlen(A)$ be the length of a sequence A (number of elements in A) then temporal scale S_T is defined as:

Algorithm 1 Correcting warp-path

```
procedure CorrectWarppath($W_k, R, U$)
    $P \leftarrow \{\}$
    for all $k$ do
        $p^R \leftarrow \{w^R_k\}$
        $n \leftarrow w^R_k - 1$
        while $n > w^R_{k-1}$ do
            if $d(R[n], R[w^R_k]) < threshold$ then    ▷ $d()$ is cosine distance
                insert $n$ at the start of $p^R$
                $n = n - 1$
            else
                break
            end if
        end while
        $n \leftarrow w^R_k + 1$
        while $n < w^R_{k+1}$ do
            if $d(R[n], R[w^R_k]) < threshold$ then
                append $n$ at the end of $p^R$
                $n = n + 1$
            else
                break
            end if
        end while
        $p^U \leftarrow \{w^U_k\}$
        do the same as above for $U$ and $p^U$
        append $(p^R, p^U)$ at the end of $P$
    end for
    return $P$
end procedure
```

$$S_T = \frac{\sum_{i=k-\tau}^{k} seqlen(p^U_i)}{\sum_{j=k-\tau}^{k} seqlen(p^R_j)} \tag{12}$$

If we define $\mathbb{P}_U$ and $\mathbb{P}_R$ as a set of indices in all $\mathbf{p}^R$ and $\mathbf{p}^U$ respectively, spatial scale of user input U (Eq. (7)) compared to R (Eq. (6)) is defined as:

$$S_S = \frac{\sum_{i \in \mathbb{P}_U, k-\tau \leq i \leq k} \left| \overrightarrow{u_i} \right|}{\sum_{j \in \mathbb{P}_R, k-\tau \leq j \leq k} \left| \overrightarrow{r_j} \right|} \tag{13}$$

For both S_T and S_S, τ $(1 \leq \tau \leq k)$ indicates an interval on which scales are evaluated. That is, if τ is $(k-1)$, scales are calculated on the whole gesture input (global scale) otherwise express local scale variation.

5. Evaluation

5.1 Analysis on Real-World Data

Performance under real-world conditions was evaluated using recorded gesture data. 5 subjects recorded circle shaped gestures with different styles. Each style has a distinctive combination of size and speed (slow, medium or fast for speed and small, medium or big for size). As a reference, medium speed and medium size of the gesture was recorded by each subject and scale variations were determined by double or half the size and speed of the reference. Subjects were requested to maintain style during performance as consistently as possible. Each style was recorded 10 times (total

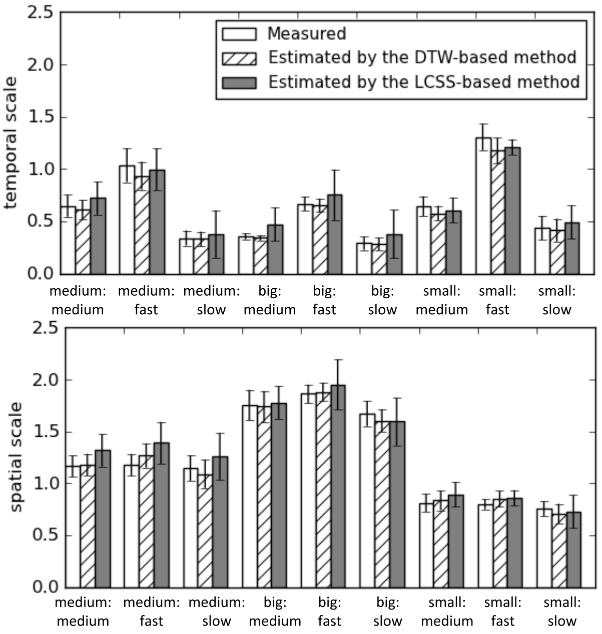


Fig. 5 An example of comparison among measured, estimated by the DTW-based method and LCSS-based method of one subject's data. Styles are described in the form of *size:speed*.

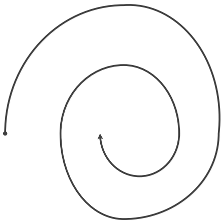


Fig. 6 A spiral gesture shape.

Table 1 Styles of spiral gestures used in simulation: the name of each style expresses changes in style. For example, a *big and fast to small and slow* spiral gesture means gesture performance starts as if a big circle drawn quickly then ends as if a small circle drawn slowly.

		temporal scale		spatial scale	
gestures	sample #	initial	final	initial	final
reference	60	1.0	1.0	1.0	1.0
big and fast to small and slow	75	2.5	0.5	2.0	0.5
big and slow to small and fast	75	0.5	2.5	2.0	0.5
small and fast to big and slow	75	2.5	0.5	0.5	2.0
small and slow to big and fast	75	0.5	2.5	0.5	2.0

90 gesture data per subject).

Unlike synthesized gestures, it is very difficult to measure the actual scales of recorded data since it contains many human errors—for example, one subject could not perform consistently even for the same style. Thus measured scale (trajectory length and performance time of raw data) should not be interpreted as ground truth. Figure 5 shows the estimation result of one subject. Although exact accuracy cannot be given by these graphs, we found consistency in measured scales and estimated scales; for instance, big and fast gestures can be grouped together or differentiated by other styles.

For all recorded gestures, standard deviation of trajectory length and performance time was 9.08% and 13.78% in average, respectively. Roughly, this can be considered as noise in gesture data under real-world conditions, which is caused by inconsistency in user performance and sensor noise.

5.2 Simulation on Synthesized Gesture Data

The accuracy of our algorithm was evaluated with synthesized gesture sequences. To assess local scale changes, a spiral shape was chosen for a target gesture class. It is a simple but practical form to be expressed easily by users since it can be thought of drawing two circles with different scales (Fig. 6). Reference data and four input gestures with different styles were created (Table 1). Also, Gaussian noise from 1% to 17% was added to input gestures and an interval for scale estimation was 10 ($\tau = 10$). For each gesture style and noise level, simulation was conducted 100 times and the results of simulation were averaged. Our algorithm was implemented with C++ and Python without optimization and on a Core i5-2540M machine (2.5 GHz, 4 GB RAM) the execution time of gesture progression matching and scale estimation for each new input sample was under 1 ms, which means that computational efficiency of the algorithm is sufficient for real-time applications.

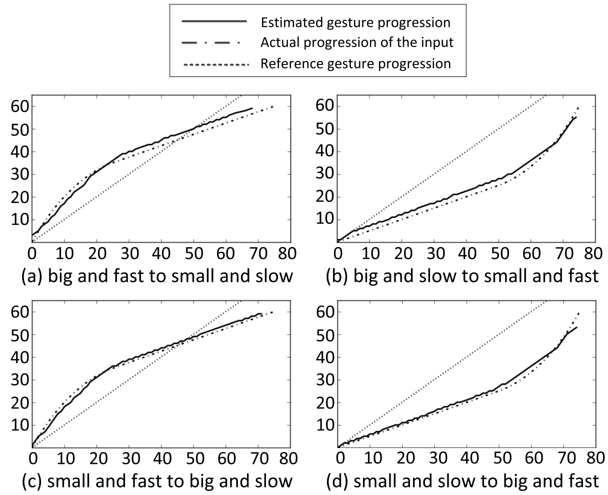
Gesture progression matching

The accuracy of progression matching (*gesture synchronization*) was assessed by comparing estimated progression to the actual progression of input gestures. Figure 7 shows the results of the overall comparison (5% of noise). For all input gestures, the estimated progression resembles the actual progression and stable through the progression. Also, average matching error (*actual progression index* − *estimated index*) under various noise levels was evaluated (Table 2). The results show that our algorithm can estimate gesture progression quite reliably under the noise levels we are assuming in Sect. 5.1. These results are comparable with the approach of Bevilacqua et al. [30], which is more susceptible to scale changes and noise.

Scale estimation

Changes in temporal and spatial scale were estimated along with gesture progression. For estimating local scale changes, scales were computed on small time interval (10 samples) and for the stability of estimation, scales are calculated only after samples are gathered more than the interval size.

Figure 8 shows the estimated and actual scale along with gesture progression when 5% of noise is added. The overall changes of the estimated scale are quite similar to the actual scale changes for all input gestures. However, the estimated scales seem like they are lagging behind and are smaller than the actual scales where scales changes unevenly (around 20th sample frame in (a) & (c) and 50th sam-



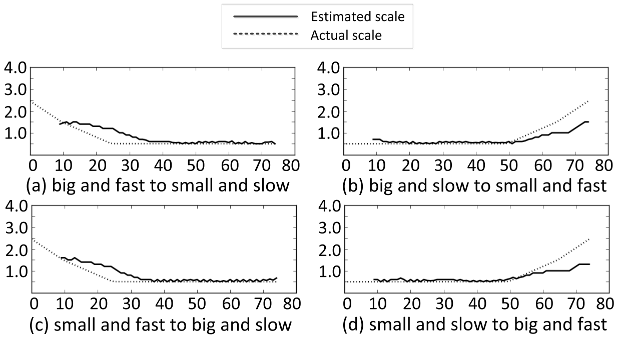
(a) big and fast to small and slow (b) big and slow to small and fast

(c) small and fast to big and slow (d) small and slow to big and fast

Fig. 7 Comparison between actual progression and estimated progression under 5% of noise. The horizontal axis is the progression of input gesture (sample index) and the vertical axis is the corresponding indices of the reference gesture. Slopes indicate progression rate in comparison to the reference. If the slope for a certain interval is greater than 1.0, it means the user input is being performed faster than the reference and vice versa.

Table 2 Average errors of progression matching in samples (standard deviation in parentheses). Gesture styles are: (a) big and fast to small and slow, (b) big and slow to small and fast, (c) small and fast to big and slow, (d) small and slow to big and fast.

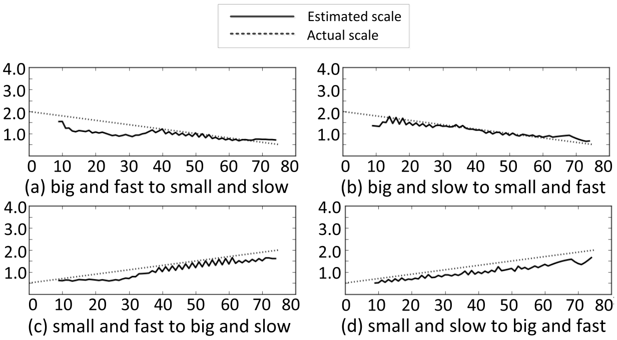
noise level (%)	gesture styles			
	(a)	(b)	(c)	(d)
1	2.30 (0.80)	2.21 (0.69)	1.33 (0.56)	1.32 (0.82)
3	2.30 (0.80)	2.20 (0.69)	1.34 (0.56)	1.31 (0.83)
5	2.30 (0.81)	2.20 (0.70)	1.33 (0.56)	1.33 (0.83)
7	2.32 (0.84)	2.20 (0.70)	1.34 (0.57)	1.35 (0.83)
9	2.29 (0.83)	2.21 (0.71)	1.34 (0.57)	1.36 (0.82)
11	2.33 (0.87)	2.20 (0.71)	1.34 (0.58)	1.38 (0.83)
13	2.32 (0.87)	2.22 (0.72)	1.34 (0.58)	1.39 (0.82)
15	2.31 (0.89)	2.22 (0.73)	1.35 (0.58)	1.44 (0.82)
17	2.31 (0.90)	2.22 (0.75)	1.36 (0.59)	1.44 (0.84)



(a) big and fast to small and slow (b) big and slow to small and fast

(c) small and fast to big and slow (d) small and slow to big and fast

Fig. 8 Evaluation of temporal scale estimation (5% noise). The horizontal axis is progression of input gesture (sample index) and the vertical axis represents temporal scale at the corresponding sample frame.



(a) big and fast to small and slow (b) big and slow to small and fast

(c) small and fast to big and slow (d) small and slow to big and fast

Fig. 9 Evaluation of spatial scale estimation (5% noise). The horizontal axis is the progression of input gesture (sample index) and the vertical axis represents spatial scale at the corresponding sample frame.

Table 3 Average errors in temporal scale estimation (standard deviations in parentheses). Gesture styles are: (a) big and fast to small and slow, (b) big and slow to small and fast, (c) small and fast to big and slow, (d) small and slow to big and fast.

noise level (%)	gesture styles			
	(a)	(b)	(c)	(d)
1	32.59 (32.38)	20.50 (12.48)	27.53 (29.06)	19.00 (12.20)
3	32.05 (32.69)	20.47 (12.56)	27.36 (29.25)	18.98 (12.31)
5	32.07 (32.98)	20.49 (12.83)	27.25 (29.33)	19.06 (12.52)
7	32.08 (32.93)	20.52 (13.00)	27.23 (29.40)	19.24 (12.85)
9	31.99 (33.06)	20.54 (13.09)	27.21 (29.59)	19.26 (13.57)
11	32.43 (33.06)	20.60 (13.23)	27.20 (29.59)	19.73 (14.00)
13	32.69 (33.05)	20.82 (13.42)	27.35 (29.71)	20.01 (14.35)
15	33.15 (33.01)	21.00 (13.54)	27.46 (29.82)	20.99 (14.66)
17	33.64 (33.29)	21.03 (13.74)	27.54 (30.23)	21.45 (14.56)

Table 4 Average errors in spatial scale estimation (standard deviations in parentheses). Gesture styles are: (a) big and fast to small and slow, (b) big and slow to small and fast, (c) small and fast to big and slow, (d) small and slow to big and fast.

noise level (%)	gesture styles			
	(a)	(b)	(c)	(d)
1	17.90 (12.87)	11.27 (10.27)	19.74 (8.91)	21.89 (5.48)
3	18.23 (12.81)	11.38 (10.21)	19.60 (8.98)	21.92 (5.78)
5	18.45 (12.88)	11.67 (10.13)	19.55 (8.98)	21.83 (6.30)
7	19.19 (12.54)	11.88 (10.13)	19.54 (8.98)	21.79 (6.96)
9	19.70 (12.50)	12.08 (10.00)	19.52 (9.09)	21.83 (8.17)
11	20.46 (12.94)	12.32 (10.08)	19.31 (9.33)	21.92 (8.91)
13	21.71 (12.69)	12.84 (10.31)	19.48 (9.42)	21.62 (9.31)
15	22.29 (13.22)	13.08 (10.23)	19.47 (9.63)	22.10 (10.42)
17	22.89 (13.10)	13.56 (10.33)	19.67 (9.95)	22.09 (10.56)

ple frame in (b) & (d)). We think the phenomenon is caused by: 1) estimation intervals consisting of past data; 2) scales are averaged on the interval. This indicates that estimation intervals should be carefully determined for accuracy and stability.

Results of spatial scale estimation are shown in Fig. 9 (also with 5% of noise). Estimated spatial scales are also quite close to the actual scales overall. However, where the input gesture is being performed faster than the reference, it seems estimation error is increased (like the initial parts of (a)). Average errors on temporal and spatial scale are listed in Table 3 and 4 respectively. The overall results show that the estimation error does not affected significantly by the noise levels we are assuming as real-world conditions. The average errors are similar or slightly lowered compared to our previous work [32]. Taking into account that Gaussian noise was not added to input gestures in the previous evaluation, this implies an increase in the accuracy of the current method. Also, decreased standard deviation (more than 80% in the worst case in the previous work) proves improved stability.

6. Conclusion and Future Work

In this paper, we presented a method for scale estimation from continuous gesture input. *Gesture synchronization* is introduced for matching gesture progression between input and a reference. Temporal and spatial scales are estimated based on progression matching. By adopting ideas of *turning angle representation*, scale-aware gesture representation is defined and an LCSS-based similarity measure is used to compare temporally mismatching gestures. Evaluation with synthesized data and recorded gestures shows the accuracy and reliability of our algorithm under noisy condition, and applicability for real-world situations.

One limitation of this work is that we assume a target gesture type is known before starting estimation. In reality, gesture types are usually unknown before gestures are performed and recognized. Therefore, it is necessary to carry out gesture recognition and estimation at the same time. However, contradicting characteristics between recognition and estimation makes this problem more difficult than thought—gesture recognition needs as much data as possible for accuracy but real-time continuous scale estimation has to be performed on incomplete partial data. Dividing gestures into a preparation stage [33] (for recognition) and a performance stage (for estimation) or using *gesture graph* [31] may be candidate solutions.

In our future work, we are planning to extend our current work to recognize the expressiveness of human motions for controlling animation characters. Expressiveness of motions could be described in size, speed, and other factors [2]. After the relationship is modeled, it may be possible to recognize motions at the affective level such as happy or sad [4], [14], [15]. We expect that our approach will enable us to control expressive animations more naturally and intuitively, which is one of the important issues in computer animation.

References

[1] S. Mitra and T. Acharya, "Gesture recognition: A survey," IEEE Trans. Syst. Man Cybern., C, Appl. Rev., vol.37, no.3, pp.311–324, 2007.

[2] P.E. Gallaher, "Individual differences in nonverbal behavior: Dimensions of style," J. Personality and Social Psychology, vol.63, no.1, pp.133–145, 1992.

[3] H.G. Wallbott and K.R. Scherer, "Cues and channels in emotion recognition," J. Personality and Social Psychology, vol.51, no.4, pp.690–699, 1986.

[4] D. Chi, M. Costa, L. Zhao, and N. Badler, "The emote model for effort and shape," Proc. 27th Annual Conference on Computer Graphics and Interactive Techniques, pp.173–182, ACM Press/Addison-Wesley Publishing, 2000.

[5] R.C. Veltkamp, "Shape matching: Similarity measures and algorithms," International Conference on, Shape Modeling and Applications, p.0188, IEEE Computer Society, 2001.

[6] E. Keogh, S. Chu, D. Hart, and M. Pazzani, "An online algorithm for segmenting time series," ICDM, p.289, IEEE Computer Society, 2001.

[7] H. Junker, O. Amft, P. Lukowicz, and G. Tröster, "Gesture spotting with body-worn inertial sensors to detect user activities," Pattern Recognit., vol.41, no.6, pp.2010–2024, 2008.

[8] T. Stiefmeier, D. Roggen, and G. Tröster, "Gestures are strings: Efficient online gesture spotting and classification using string matching," Proc. ICST 2nd International Conference on Body Area Networks, pp.1–8, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2007.

[9] R.H. Liang and M. Ouhyoung, "A real-time continuous gesture recognition system for sign language," Face and Gesture Recognition, pp.558–567, 1998.

[10] H.K. Lee and J.H. Kim, "An hmm-based threshold model approach for gesture recognition," IEEE Trans. Pattern Anal. Mach. Intell., vol.21, no.10, pp.961–973, 1999.

[11] A.D. Wilson and A.F. Bobick, "Parametric hidden Markov models for gesture recognition," IEEE Trans. Pattern Anal. Mach. Intell., vol.21, no.9, pp.884–900, 2002.

[12] D. Herzog, A. Ude, and V. Kruger, "Motion imitation and recognition using parametric hidden markov models," Humanoid Robots, 2008. Humanoids 2008. 8th IEEE-RAS International Conference on, pp.339–346, IEEE, 2009.

[13] C. Appert and O. Bau, "Scale detection for a priori gesture recognition," Proc. 28th International Conference on Human Factors in Computing Systems, pp.879–882, ACM, 2010.

[14] A. Camurri, B. Mazzarino, and G. Volpe, "Analysis of expressive gesture: The eyesweb expressive gesture processing library," Gesture-Based Communication in Human-Computer Interaction, pp.469–470, 2004.

[15] A. Camurri, B. Mazzarino, M. Ricchetti, R. Timmers, and G. Volpe, "Multimodal analysis of expressive gesture in music and dance performances," Gesture-Based Communication in Human-Computer Interaction, pp.357–358, 2004.

[16] M. Rehm, N. Bee, and E. André, "Wave like an egyptian: Accelerometer based gesture recognition for culture specific interactions," Proc. 22nd British HCI Group Annual Conference on HCI 2008: People and Computers XXII: Culture, Creativity, Interaction - Volume 1, pp.13–22, British Computer Society, 2008.

[17] M. Thorne, D. Burke, and M. van de Panne, "Motion doodles: An interface for sketching character motion," ACM SIGGRAPH 2007 courses, p.24, ACM, 2007.

[18] T. Shiratori and J.K. Hodgins, "Accelerometer-based user interfaces for the control of a physically simulated character," ACM Trans. Graphics (TOG), pp.123:1–123:9, ACM, Dec. 2008. ACM ID: 1409076.

[19] S.D. Cohen and L.J. Guibas, "Partial matching of planar polylines under similarity transformations," Proc. 8th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '97, pp.777–786, Society for Industrial and Applied Mathematics, 1997. ACM ID: 314445.

[20] M. Rusiñol, P. Dosch, and J. Lladós, "Boundary shape recognition using accumulated length and angle information," Pattern Recognition and Image Analysis, pp.210–217, 2007.

[21] D.M. Gavrila and L.S. Davis, "Towards 3-d model-based tracking and recognition of human movement: A multi-view approach," International Workshop on Automatic Face-and Gesture-Recognition, pp.272–277, Citeseer, 1995.

[22] T.M. Rath and R. Manmatha, "Word image matching using dynamic time warping," 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings, 2003.

[23] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," Readings in Speech Recognition,

pp.159–165, 1990.

[24] M. Vlachos, D. Gunopulos, and G. Das, "Rotation invariant distance measures for trajectories," Proc. 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp.707–712, ACM, 2004.

[25] M. Vlachos, D. Gunopoulos, and G. Kollios, "Discovering similar multidimensional trajectories," ICDE, p.0673, IEEE Computer Society, 2002.

[26] M. Vlachos, M. Hadjieleftheriou, D. Gunopulos, and E. Keogh, "Indexing multi-dimensional time-series with support for multiple distance measures," Proc. 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '03, pp.216–225, ACM, 2003. ACM ID: 956777.

[27] A. Croitoru, P. Agouris, and A. Stefanidis, "3d trajectory matching by pose normalization," Proc. 13th Annual ACM International Workshop on Geographic Information Systems, GIS '05, pp.153–162, ACM, 2005. ACM ID: 1097087.

[28] D. Buzan, S. Sclaroff, and G. Kollios, "Extraction and clustering of motion trajectories in video," Pattern Recognit., 2004. ICPR 2004. Proc. 17th International Conference on, vol.2, pp.521–524, IEEE.

[29] L. Chen, M.T. Zsu, and V. Oria, "Robust and fast similarity search for moving object trajectories," Proc. 2005 ACM SIGMOD International Conference on Management of Data, pp.491–502, ACM, 2005.

[30] F. Bevilacqua, B. Zamborlin, A. Sypniewski, N. Schnell, F. Gudy, and N. Rasamimanana, "Continuous realtime gesture following and recognition," Gesture in Embodied Communication and Human-Computer Interaction, pp.73–84, 2010.

[31] A. Mori, S. Uchida, R. Kurazume, R. Taniguchi, T. Hasegawa, and H. Sakoe, "Early recognition and prediction of gestures," Pattern Recognition, 2006. ICPR 2006. 18th International Conference on, pp.560–563, IEEE, 2006.

[32] W. Kim and H. Kuzuoka, "A scale-aware gestural interface for video games," Proc. NICOGRAPH INTERNATIONAL 2011, 2011.

[33] J. Lasseter, "Principles of traditional animation applied to 3d computer animation," SIGGRAPH Comput. Graph., vol.21, no.4, pp.35–44, 1987.

Woosuk Kim received his B.S. and M.S. in electrical & computer engineering, in 2001 and 2003, from Hanyang University, Korea. From 2003 to 2007, he was a member of engineering staff at Electronics and Telecommunications Research Institute, Korea. Since 2007, he is in the doctoral program of Intelligent Interaction Technologies at Tsukuba University, Japan.

Hideaki Kuzuoka received his Ph.D. degree in Information Engineering in 1992 from the University of Tokyo. He is currently a Professor of in the Faculty of Engineering, Information and Systems at University of Tsukuba. His current research interest includes CSCW, human-robot interaction, and Virtual Reality.

Kenji Suzuki is currently an Assistant Professor at the Center for Cybernics Research, and also Principle Investigator of Artificial Intelligence Laboratory, University of Tsukuba, Japan. He received the B.S. in Physics, M.E. and Ph.D. in Pure and Applied Physics from Waseda University, Tokyo, Japan, in 1997, 2000 and 2003 respectively. Prior to joining the University of Tsukuba, he was a Research Associate at the Dept. of Applied Physics, Waseda University, Japan. He was also a visiting researcher at the Laboratory of Physiology of Perception and Action, College de France in Paris, and the Laboratory of Musical Information, University of Genoa, Italy. His research interests include Assistive and Cognitive Robotics, Humanoid Robotics, Augmented Human Technology, Biosignal Processing, and Social Playware.