

# Vector-based, Structure Preserving Stroke Gesture Recognition

Nathan Magrofuoco<sup>1</sup>, Paolo Roselli<sup>2</sup>, J.L. Pérez Medina<sup>1,3</sup>, Jean Vanderdonckt<sup>1</sup>, Santiago Villarreal<sup>1</sup>

<sup>1</sup> Université catholique de Louvain (UCL). Louvain-la-Neuve, Belgium

<sup>2</sup> Università degli Studi di Roma “Tor Vergata”. Rome, Italy

<sup>3</sup> Intelligent & Interactive Systems Lab, Universidad de las Américas. Quito, Ecuador

**Abstract**—Unistroke and multistroke gesture recognizers have always striven to reach some robustness with respect to all variations encountered when people issue gestures by hand on touch surfaces or with sensing devices. For this purpose, successful stroke recognizers rely on a gesture recognition algorithm that satisfies a series of invariance properties such as: stroke-order invariance, stroke-number invariance, stroke-direction invariance, position, scale, and rotation invariance. Before initiating any recognition activity, these algorithms ensure these properties by performing several pre-processing operations. These operations induce an additional computational cost to the recognition process, as well as a potential error bias. To cope with this problem, we introduce an algorithm that ensures all these properties analytically instead of statistically based on a vector algebra. Instead of points, the recognition algorithm works on vectors between vectors. We demonstrate that this approach not eliminates the need for these pre-processing operations but also satisfies an entire structure-preserving transformation.

## I. INTRODUCTION

Gesture user interfaces [4], [6], [28] are User Interfaces (UIs) fostering a natural interaction by capturing gestures issued by the user and by letting the system to recognize them and turn them into commands. Such UIs are applicable to a wide range of activities, such as menu item selection by gesture [2], hand recognition [13], diagram sketching [7], electronic commerce [3], handwriting [10].

Recognizers exploit a large palette of techniques to recognize stroke gestures [28], such as machine learning [6], data mining [5], template-based matching [23], pattern recognition [11], or Nearest Neighbor Classification (NCC) [1], [9], [12], [13], [16], [19], [25], [24], [27] due to several demonstrated advantages: high recognition rate, low execution time, low memory consumption, reasonable computational complexity, and understandable geometric interpretation.

When a stroke is acquired by an end user on a particular device (e.g., a smartphone, a trackpad, a ring device), it is usually subject to several potential variations which could affect its recognition: variations due to the end user, the input device, the output device, the underlying operating system and its software. Any stroke gesture recognizer is perpetually challenged by coping with these variations and address them by ensuring some properties of invariance, like stroke-order invariance, stroke-number invariance, stroke-direction invariance, position, scale, and rotation invariance [11]. To establish and preserve these properties, recognizers perform

several pre-processing steps (e.g., re-sampling, re-rotation, normalization) before any recognition activity is actually launched, thus consuming some resources. A vector-based recognizer escapes from performing these pre-processing steps, thus saving important resources. Working directly with vectors enables us to be released from the physical coordinates space and calculations, which mix points, lines, angles, surfaces, volumes. To discuss these aspects, this paper proves some invariance properties, leading to a structure-preserving recognizer that is beyond articulation-invariant recognizers. This paper extends [22] as follows: it focuses on structure preservation by elaborating on invariance properties, by adding isotropicity, it provides the mathematical demonstrations of invariance properties, it expresses Rubine features in terms of vectors, and it introduces shape distance for unoriented and oriented curves.

The remainder of this paper is structured as follows: Section 2 discusses related work with a focus on invariance properties; Section 3 introduces a vector-based approach for stroke gesture recognition; Section 4 demonstrates how structure-preserving capability is achieved by ensuring invariance properties algebraically instead of statistically or heuristically; Section 5 concludes this work.

## II. RELATED WORK

A stroke gesture recognizer is said to be *invariant* if it remains robust, stable, or independent enough with respect to variations of a gesture to be recognized. Such variations can be classified into two categories: a *global variation* concerns a transformation applied to a gesture considered as a whole (e.g., a scaling, a dilatation) while a *local variation* only concerns a specific part of a gesture (e.g., cropping).

A first invariance property is *training set-invariance*: when the training set is modified (e.g., for adding a new gesture, for removing, or modifying an existing one), the algorithm remains unmodified. For example, *synthetic gestures* [14] enrich a training set without requiring more participants to provide additional samples.

*Sampling invariance* refers to the independence of a recognizer with respect to the amount of points [11], and therefore strokes between these points. Experiments show that 8 points are sufficient to recognize a sample stroke, but best results are obtained with 32 points [25].

*Direction invariance* [11], or *isotropicity*, refers to the independence of a recognizer with respect to the direction and ordering of stroke composing a gesture. *Time invariance*

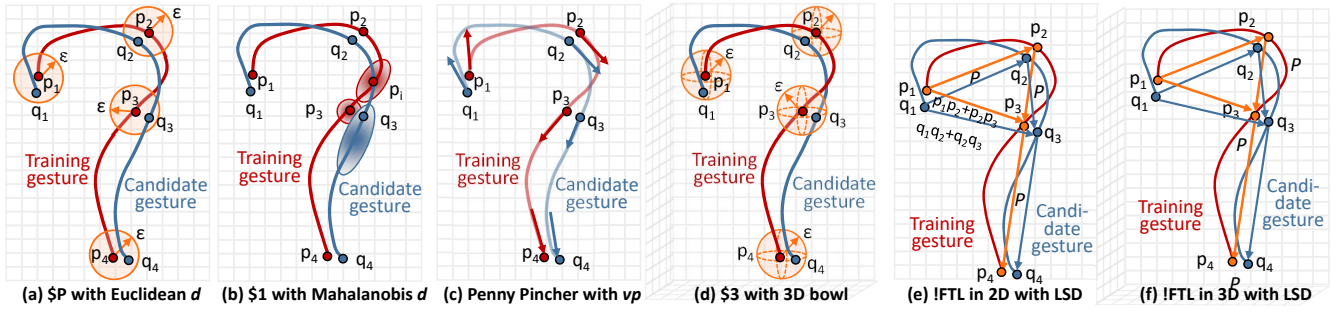


Fig. 1. Geometric interpretation of gesture recognition of a candidate gesture against a reference gesture: (a) 2D distance in SP, (b) 3D bowl of center  $p_i$  and radius  $\epsilon$  in \$3, (c) 2D vector-based LSD, (d) 3D vector-based LSD.

[23] refers to the independence of a recognizer with respect to the time required by a end user to issue a gesture.

*Position invariance* refers to the independence of a recognizer to recognize a gesture wherever it is located on an interaction surface or in the air. Signer [18], imposed that a gesture should be issued in a given bounding box on a certain surface because recognition is conditioned by a partition of this space into regions.

*Scale invariance* refers to the independence of a recognizer with respect to size and scale of the gesture. People tend to issue larger gestures on a large surface [26]. UI sketches are larger on a tabletop than on a graphical tablet even if they represent the same concept [7]. Gesture size could also convey a particular meaning: a small, medium, or large arrow expresses a slow, medium, or fast navigation movement.

*Rotation invariance* refers to the independence of a recognizer to recognize the same gesture whatever its orientation is. People sitting around a circular tabletop need all orientations possible since no reference position is privileged. For instance, PennyPincher [19] in its early version was not rotation-invariant while its successor was [20]. Protractor3D [13] supports 3D gestures in a rotation-invariant way.

*Articulation invariance* groups the above properties together to refer to a recognizer that does not depend on how end users may produce gestures to give results.

*Reflection invariance* refers to the independence of a recognizer to recognize the same gesture whatever the face of the interactive surface is, such as on a mirror or a transparent window. Since end users are not generally aware of the influence of their location with respect to the display, they prefer to forget it: two users located on each part of a transparent display collaborate regardless the reflection.

*Projection invariance* expresses the same property in space: whether a gesture is issued in the vertical, horizontal, or sagittal plane, its recognition remains the same. Projection invariance is more complicated to satisfy because points acquired in a particular system of coordinates may not mesh well with a projection. Several pre-processing steps should be performed: resampling or reparameterization, edgification, normalization, rotation to a reference point.

These pre-processing steps satisfy their corresponding properties, but negatively affect the overall performance, its complexity, and could introduce some approximations and/or error rounding due to these calculations that have nothing to do with the recognition process.

### III. VECTOR-BASED GESTURE RECOGNITION

#### A. Initial Definition

To characterize structure preservation, we refer to *isometricity* as the property of a gesture set to hold a set of  $n$ -equally distanced points:  $\forall i \in \overline{n-1}, d(p_i, p_{i+1}) = \text{constant}$ . A representative example exist when  $\|p_{i+1} - p_i\| = \frac{1}{n-1} \sum_{i=1}^n \|p_{i+1} - p_i\|$ . We hereby refer to *isochronicity* as the property of a gesture set to hold a set of  $n$  equally-timestamped points, i.e.,  $\forall i \in \overline{n-1}, \|t_i - t_{i+1}\| = \text{constant}$ , e.g.,  $\|t_{i+1} - t_i\| = \frac{1}{n-1} \sum_{i=1}^n \|(t_{i+1} - t_i)\|$ . We refer to *isotropicity* as the property of a gesture set to hold a set of stroke having the same orientation, such as left-to-right (L2R). We also define *isoparametrization* as the property of two or more gestures/sets to contain the same amount of points, i.e.  $\forall G = \{p_i\}_{i=1, \dots, n}, H = \{q_j\}_{j=1, \dots, m} : m=n$ . Two gesture sets could be isoparametrized whether they are isometric or not, isochronic or not, isotropic or not.

A gesture expresses a motion between an initial point and a final point based on features such as position, scale, direction, curvature, pressure, tangential acceleration, to be extracted and classified [5], [21]. A point-based representation of a gesture has the advantage of significantly simplifying a gesture to a series of points, thus reducing the gesture recognition to a comparison of two series of points. This approach works well for static shapes, like symbols, letters, simple commands, where shape is more important than motion.

#### B. Definition of a Basic Gesture

The continuous trace of two consecutive non-trivial translations of a point will be called *basic gesture*. A basic gesture in a finite dimensional affine space can then be formalized by a ordered couple  $(\vec{u}, \vec{v})$  of two non-zero free vectors  $\vec{u} \in \mathbb{R}^n$  and  $\vec{v} \in \mathbb{R}^n$ . A basic gesture  $(\vec{u}, \vec{v})$  generates a precise oriented triangle, whose third oriented side is the free vector  $-(\vec{u} + \vec{v})$ . However, a well precise (possibly non trivial) triangle, having points  $A, B$  and  $C$  as distinct vertices can be generated by six possibly different basic gestures. If we denote  $\vec{a} = C - B$ ,  $\vec{b} = A - C$ ,  $\vec{c} = B - A$ , then we have six basic gestures:  $(\vec{c}, \vec{a})$ ,  $(\vec{a}, \vec{b})$ ,  $(\vec{b}, \vec{c})$ ,  $(-\vec{a}, -\vec{c})$ ,  $(-\vec{c}, -\vec{b})$ , and  $(-\vec{b}, -\vec{a})$ , respectively. In particular, basic gestures in an affine plane correspond to ordered couples  $(\vec{u}, \vec{v})$  of two non-zero free vectors  $\vec{u} = (u_1, u_2) \in \mathbb{R}^2$  and  $\vec{v} = (v_1, v_2) \in \mathbb{R}^2$ . There is a one-to-one correspondence between vectors in  $\mathbb{R}^2$  and complex numbers in  $\mathbb{C}$ ; more precisely, to each free vector  $\vec{x} = (x_1, x_2) \in \mathbb{R}^2$  corresponds the complex number

$\mathbf{x} = x_1 + ix_2 \in \mathbb{C}$ , and vice versa ( $i$  is the imaginary unit such that  $i^2 = -1$ ). The shape of an ordered triangle, traced by a basic gesture  $(\vec{u}, \vec{v})$ , can be encoded by the complex number [15] obtained as the quotient  $\frac{\mathbf{u}}{\mathbf{v}} \in \mathbb{C}$  of the complex numbers  $\mathbf{u}, \mathbf{v} \in \mathbb{C}$  corresponding to vectors  $\vec{u}, \vec{v} \in \mathbb{R}^2$  of the basic gesture  $(\vec{u}, \vec{v})$ . Every triangle can be characterized by its *shape* [15].

### C. Vector-based Features

Gesture features are traditionally defined based on Cartesian coordinates. A transformation can convert them into any other system of coordinates, like polar, or spherical. Conversely, when a gesture is captured by a non-Cartesian device, a transformation into Cartesian coordinates should be applied before computing classical features, such as Rubine's features [17]. Instead, our vector-based definition re-expresses these features in terms of vectors that are independent of any system coordinates. We vectorized Rubine's features [17] into a red formula as follows:

Notations:  $\forall (x_p, y_p) = \vec{u}_p$  with  $0 \leq p \leq P-1$

$$\begin{aligned} \Delta x_p &= x_{p+1} - x_p, \Delta y_p = y_{p+1} - y_p \\ (\Delta x_p, \Delta y_p) &= \Delta \vec{u}_p = \vec{u}_{p+1} - \vec{u}_p \text{ and } \Delta t_p = t_{p+1} - t_p \\ \theta_p &= \arctan \frac{\Delta x_p \Delta y_{p-1} - \Delta x_{p-1} \Delta y_p}{\Delta x_p \Delta x_{p-1} + \Delta y_p \Delta y_{p-1}} \\ &= \arctan \frac{(\Delta \vec{u}_p \wedge \Delta \vec{u}_{p-1}) \cdot (\vec{e}_1 \wedge \vec{e}_2)}{\Delta \vec{u}_p \cdot \Delta \vec{u}_{p-1}} \end{aligned} \quad (1)$$

Let  $\{\vec{e}_1, \vec{e}_2\}$  be an orthonormal basis in the two-dimensional Euclidean space  $\mathbb{E}_2$ . Then, Rubine's features become:

$$\begin{aligned} f_1 &= \cos \alpha = \frac{x_2 - x_0}{\sqrt{(x_2 - x_0)^2 + (y_2 - y_0)^2}} = \frac{(\vec{u}_2 - \vec{u}_0) \cdot \vec{e}_1}{|\vec{u}_2 - \vec{u}_0|} \\ f_2 &= \sin \alpha = \frac{y_2 - y_0}{\sqrt{(x_2 - x_0)^2 + (y_2 - y_0)^2}} = \frac{(\vec{u}_2 - \vec{u}_0) \cdot \vec{e}_2}{|\vec{u}_2 - \vec{u}_0|} \\ f_5 &= \sqrt{(x_{P-1} - x_0)^2 + (y_{P-1} - y_0)^2} = |\vec{u}_{P-1} - \vec{u}_0| \\ f_6 &= \cos \beta = \frac{x_{P-1} - x_0}{\sqrt{(x_{P-1} - x_0)^2 + (y_{P-1} - y_0)^2}} = \frac{(\vec{u}_{P-1} - \vec{u}_0) \cdot \vec{e}_1}{|\vec{u}_{P-1} - \vec{u}_0|} \\ f_7 &= \sin \beta = \frac{y_{P-1} - y_0}{\sqrt{(x_{P-1} - x_0)^2 + (y_{P-1} - y_0)^2}} = \frac{(\vec{u}_{P-1} - \vec{u}_0) \cdot \vec{e}_2}{|\vec{u}_{P-1} - \vec{u}_0|} \\ f_8 &= \sum_{p=0}^{P-2} \sqrt{\Delta x_p^2 + \Delta y_p^2} = \sum_{p=0}^{P-2} |\vec{u}_{p+1} - \vec{u}_p| = \sum_{p=0}^{P-2} |\Delta \vec{u}_p| \\ f_9 &= \sum_{p=1}^{P-2} \theta_p = \arctan \frac{(\Delta \vec{u}_{p-1} \wedge \Delta \vec{u}_1) \cdot (\vec{e}_1 \wedge \vec{e}_2)}{\Delta \vec{u}_{p-1} \cdot \Delta \vec{u}_1} = \gamma \\ f_{10} &= \sum_{p=1}^{P-2} |\theta_p| \text{ and } f_{11} = \sum_{p=1}^{P-2} |\theta_p|^2 \text{ and } f_{13} = t_{P-1} - t_0 \\ f_{12} &= \max_{p=0}^{P-2} \frac{\Delta x_p^2 + \Delta y_p^2}{\Delta t_p^2} = \max_{p=0}^{P-2} \frac{|\Delta \vec{u}_p|^2}{\Delta t_p^2} = \max_{p=0}^{P-2} \left| \frac{1}{\Delta t_p} \Delta \vec{u}_p \right|^2 \end{aligned} \quad (2)$$

### D. Similarity Ratio

Quotient  $\frac{\mathbf{u}}{\mathbf{v}} \in \mathbb{C}$  is defined as the *similarity ratio* of the basic gesture  $(\vec{u}, \vec{v})$ . Let us recall the properties and correspondences between the metrics in  $\mathbb{R}^2$  and  $\mathbb{C}$ : the norm of a free vector  $\vec{x} \in \mathbb{R}^2$  and the modulus of its corresponding complex number  $\mathbf{x} \in \mathbb{C}$  coincide

$$|\vec{x}| = \sqrt{\vec{x} \cdot \vec{x}} = \sqrt{(x_1)^2 + (x_2)^2} = \sqrt{\mathbf{x} \mathbf{x}^*} = |\mathbf{x}|_{\mathbb{C}} \text{ where } (3)$$

- $\vec{x} \cdot \vec{y} = (x_1 y_1) + (x_2 y_2)$  is the scalar product between  $\vec{x} = (x_1, x_2) \in \mathbb{R}^2$  and  $\vec{y} = (y_1, y_2) \in \mathbb{R}^2$ ,
- $\mathbf{x}^* = x_1 - ix_2 \in \mathbb{C}$  is the complex conjugate of  $\mathbf{x} = x_1 + ix_2 \in \mathbb{C}$ .

Thus, also the distance between two vectors  $\vec{x}, \vec{y} \in \mathbb{R}^2$  and the distance between the two corresponding complex numbers  $\mathbf{x}, \mathbf{y} \in \mathbb{C}$  coincide:  $|\vec{x} - \vec{y}| = |x - y|_{\mathbb{C}}$ , that is,  $\mathbb{R}^2$  and  $\mathbb{C}$  are isometric.

### E. Local Shape Distance between Gestures

We define the dissimilarity between two basic gestures  $(\vec{a}, \vec{b})$  and  $(\vec{u}, \vec{v})$ , by the *Local Shape distance* [22], denoted by the symbol  $LSD((\vec{a}, \vec{b}), (\vec{u}, \vec{v}))$ , is defined as the Euclidean distance between the similarity ratios of the basic gestures  $(\vec{a}, \vec{b})$  and  $(\vec{u}, \vec{v})$ , i.e.,

$$LSD((\vec{a}, \vec{b}), (\vec{u}, \vec{v})) = \left| \frac{\mathbf{a}}{\mathbf{b}} - \frac{\mathbf{u}}{\mathbf{v}} \right|_{\mathbb{C}} = \sqrt{\frac{|\vec{a}|^2 |\vec{v}|^2 + |\vec{b}|^2 |\vec{u}|^2 - 2[(\vec{a} \cdot \vec{b})(\vec{u} \cdot \vec{v}) - (\vec{a} \cdot \vec{v})(\vec{b} \cdot \vec{u}) + (\vec{a} \cdot \vec{u})(\vec{b} \cdot \vec{v})]}{|\vec{b}| |\vec{v}|}} \quad (4)$$

The Local Shape Distance also exhibits a geometric interpretation: a basic gesture correspond to a triangle whose two first sides are formed by two vectors and whose last side correspond to a vector joining its origin to its destination. The Local Shape distance therefore interprets the comparison of a candidature gesture to a training gesture by computing the similarity between the two corresponding triangles. This process is repeated until the end of points is reached.

$$\begin{aligned} \left| \frac{\mathbf{a}}{\mathbf{b}} - \frac{\mathbf{u}}{\mathbf{v}} \right|_{\mathbb{C}}^2 &= \left| \frac{\mathbf{av} - \mathbf{bu}}{\mathbf{bv}} \right|_{\mathbb{C}}^2 = \frac{|\mathbf{av} - \mathbf{bu}|_{\mathbb{C}}^2}{|\mathbf{bv}|_{\mathbb{C}}^2} = \frac{(\mathbf{av} - \mathbf{bu})(\mathbf{av} - \mathbf{bu})^*}{|\mathbf{bv}|_{\mathbb{C}}^2} \\ &= \frac{(\mathbf{av} - \mathbf{bu})(\mathbf{a}^* \mathbf{v}^* - \mathbf{b}^* \mathbf{u}^*)}{|\mathbf{b}|_{\mathbb{C}}^2 |\mathbf{v}|_{\mathbb{C}}^2} = \frac{\mathbf{aa}^* \mathbf{vv}^* + \mathbf{bb}^* \mathbf{uu}^* - \mathbf{ab}^* \mathbf{vu}^* - \mathbf{a}^* \mathbf{buv}^*}{|\mathbf{b}|_{\mathbb{C}}^2 |\mathbf{v}|_{\mathbb{C}}^2} \end{aligned}$$

Let us recall that if  $\vec{x} = (x_1, x_2)$  and  $\vec{y} = (y_1, y_2)$  are two free vectors in  $\mathbb{R}^2$  and their corresponding complex numbers in  $\mathbb{C}$  are  $\mathbf{x} = x_1 + ix_2$  and  $\mathbf{y} = y_1 + iy_2$  respectively, then

$$\begin{aligned} \vec{x} \cdot \vec{y} &= x_1 y_1 + x_2 y_2 = (x_1 + ix_2)(y_1 - iy_2)(x_1 - ix_2)(y_1 + iy_2) \\ &= \mathbf{xy}^* = \mathbf{x}^* \mathbf{y}. \end{aligned}$$

$$\begin{aligned} \text{Hence, we write } \mathbf{aa}^* \mathbf{vv}^* + \mathbf{bb}^* \mathbf{uu}^* &= |\vec{a}|^2 |\vec{v}|^2 + |\vec{b}|^2 |\vec{u}|^2 \text{ and} \\ \mathbf{ab}^* \mathbf{vu}^* + \mathbf{a}^* \mathbf{buv}^* &= 2\mathbf{ab}^* \mathbf{vu}^* - \mathbf{ab}^* \mathbf{vu}^* + 2\mathbf{a}^* \mathbf{buv}^* - \mathbf{a}^* \mathbf{buv}^* \\ &= 2\mathbf{ab}^* \mathbf{vu}^* + 2\mathbf{a}^* \mathbf{ubv}^* - \mathbf{ab}^* \mathbf{u}^* \mathbf{v} - \mathbf{ab}^* \mathbf{vu}^* \\ &= 2\mathbf{ab}^* \mathbf{vu}^* + 2\mathbf{a}^* \mathbf{ubv}^* - 2\mathbf{ab}^* \mathbf{u}^* \mathbf{v} \\ &= 2(\vec{a} \cdot \vec{b})(\vec{v} \cdot \vec{u}) + 2(\vec{a} \cdot \vec{u})(\vec{b} \cdot \vec{v}) - 2(\vec{a} \cdot \vec{b})(\vec{u} \cdot \vec{v}) \end{aligned}$$

Equation (5) shows that LSD can be defined without considering complex numbers. With respect to the Euclidean Clifford [8] vector algebra  $\mathcal{C}\ell_{(2,0)}$ , associated to the Euclidean vector space  $\mathbb{R}^2$ , one can recognize that

$$LSD((\vec{a}, \vec{b}), (\vec{u}, \vec{v})) = \left| \vec{a}(\vec{b})^{-1} - \vec{u}(\vec{v})^{-1} \right|_{\mathcal{C}\ell_{(2,0)}}$$

the Euclidean distance in  $\mathcal{C}\ell_{(2,0)}$  between two ratios of vectors, where

$$\vec{x}(\vec{y})^{-1} = \frac{\vec{x} \cdot \vec{y}}{|\vec{y}|^2} + \frac{1}{|\vec{y}|^2} (\vec{x} \wedge \vec{y}) \in \mathcal{C}\ell_{(2,0)}$$

In general, the two free vectors  $\vec{u}$  and  $\vec{v}$  of a basic gesture  $(\vec{u}, \vec{v})$ , can belong to a finite  $n$ -dimensional Euclidean space  $\mathbb{R}^n$ . Moreover, the LSD defined in (5) makes sense and corresponds to the distance between the ratio multivectors in the corresponding Euclidean Clifford algebra  $\mathcal{C}\ell_{(n,0)}$ , i.e.,

$$LSD((\vec{a}, \vec{b}), (\vec{u}, \vec{v})) = \left| \vec{a}(\vec{b})^{-1} - \vec{u}(\vec{v})^{-1} \right|_{\mathcal{C}\ell_{(n,0)}}$$

#### F. Invariance Properties of the Local Shape Distance

In order to demonstrate the invariance properties of LSD, let us suppose to use function  $LSD$  to compare two basic gestures  $(\vec{a}, \vec{b})$  and  $(\vec{u}, \vec{v})$  laying on a same affine plane.

**Stroke-number invariance.** LSD is computed on a set of vectors that are either continuous (a basic gesture) or a series of continuous ones (a series of basic gestures). If no stroke exists between the ending point of a gesture stroke and the starting point of the next stroke, no vector is created and the LSD remains unaffected. Therefore, both single-stroke and multi-stroke gestures are supported. The sampling limit can be pushed to only 8 points [25], but we observe that the 32 points-sampling represents a viable compromise between recognition rate and execution time.

**Sampling invariance.** LSD is computed on vectors created from one or many series of points, provided that isoparametrization is ensured. How the points are considered and thus how vectors are defined for computing the LSD does not affect LSD. Thus, stroke-order invariance is supported.

**Direction invariance.** LSD is not symmetric: it can happen that  $LSD(\vec{a}, \vec{b}, \vec{c}, \vec{d}) \neq LSD(\vec{b}, \vec{a}, \vec{d}, \vec{c})$ . An oriented gesture can be transformed into its unoriented version and redirected into a reference vector in one direction. We want to avoid edgification. Instead, a simpler and more efficient approach is preferred: since only the denominator differs for both  $LSD(\vec{a}, \vec{b}, \vec{c}, \vec{b})$  and  $LSD(\vec{b}, \vec{a}, \vec{d}, \vec{c})$ , we compute the symetrized version which is independent of the direction:

$$LSD_{sym}(\vec{a}, \vec{b}, \vec{c}, \vec{d}) = \frac{LSD(\vec{a}, \vec{b}, \vec{c}, \vec{d}) + LSD(\vec{b}, \vec{a}, \vec{d}, \vec{c})}{2}$$

One can also observe that both  $LSD$  and  $LSD_{sym}$  depend on the lengths of vectors  $\vec{a}$ ,  $\vec{b}$ ,  $\vec{c}$ , and  $\vec{d}$ . Such sensitivity to the length of each basic gesture is valuable when the points of the corresponding ordered triangle faithfully represents the isochrone sampling of a real gesture. The sampling offered by a real-world device is rarely isochrone. Equally spaced successive sampled points, i.e., isometric points, do not correspond to equally time-spaced points of the gesture, i.e., to isochrone points. In order to reduce this potential bias, we define a LSD independent on the lengths of the vectors in the basic gestures:  $\forall(\vec{a}, \vec{b}), (\vec{c}, \vec{d})$ , the Normalized Local Shape Distance (NLSD) is defined as follows:

$$NLSD((\vec{a}, \vec{b}), (\vec{c}, \vec{d})) = LSD\left(\left(\frac{\vec{a}}{|\vec{a}|}, \frac{\vec{b}}{|\vec{b}|}\right), \left(\frac{\vec{c}}{|\vec{c}|}, \frac{\vec{d}}{|\vec{d}|}\right)\right)$$

$NLSD$  is symmetric:  $NLSD((\vec{a}, \vec{b}), (\vec{c}, \vec{d})) = NLSD((\vec{b}, \vec{a}), (\vec{d}, \vec{c}))$ .

**Position Invariance.** This is ensured by proving that a translation preserves LSD. The translation performed by a free vector  $\vec{t}$  of a basic gesture  $(\vec{a}, \vec{b})$  (corresponding to a oriented triangle of vertices  $A$ ,  $B$  and  $C$ ) produces a basic gesture  $(\vec{a}', \vec{b}')$  (corresponding to a oriented triangle of vertices  $A'=A+\vec{t}$ ,  $B'=B+\vec{t}$  and  $C'=C+\vec{t}$ ). Any translation  $\vec{t}$  does not affect the basic gesture, thanks to  $\vec{a}'=\vec{a}$  and  $\vec{b}'=\vec{b}$ .

**Scale Invariance.** This is ensured by proving that a homogeneous dilation preserves LSD.  $\forall \lambda \neq 0 \in \mathbb{R}$ , we have that

$$LSD((\lambda \vec{a}, \lambda \vec{b}), (\vec{u}, \vec{v})) = \left| \frac{\lambda \vec{a}}{\lambda \vec{b}} - \frac{\vec{u}}{\vec{v}} \right|_{\mathbb{C}} = \left| \frac{\vec{a}}{\vec{b}} - \frac{\vec{u}}{\vec{v}} \right|_{\mathbb{C}} = LSD((\vec{a}, \vec{b}), (\vec{u}, \vec{v}))$$

**Rotation Invariance.** This is ensured by proving that a rotation preserves LSD. One can note that

- 1) Every counter-clockwise rotation  $\mathcal{R}_\alpha$  in the Euclidean plane  $\mathbb{R}^2$  of a Radian angle  $\alpha \in \mathbb{R}$  corresponds to a counter-clockwise rotation  $R_\alpha$  in the isometric complex plane  $\mathbb{C}$  (of equal angle).
- 2) Recalling the Euler's formulas  $e^{i\alpha} = \cos \alpha + i \sin \alpha$ ,  $e^{i(\alpha+\beta)} = e^{i\alpha} e^{i\beta}$ , the complex number  $R_\alpha(\mathbf{x})$  can be obtained throughout the complex multiplication  $R_\alpha(\mathbf{x}) = e^{i\alpha} \mathbf{x}$ .
- 3) The ratio of two complex numbers, both rotated by a same angle, is equal to their original ratio,

$$\frac{R_\alpha(\mathbf{x})}{R_\alpha(\mathbf{y})} = \frac{e^{i\alpha} \mathbf{x}}{e^{i\alpha} \mathbf{y}} = e^{i\alpha} \mathbf{x} (e^{i\alpha} \mathbf{y})^{-1} = e^{i\alpha} \mathbf{x} e^{-i\alpha} \mathbf{y}^{-1} = \underbrace{e^{i(\alpha-\alpha)}}_{=1} \mathbf{x} \mathbf{y}^{-1} = \frac{\mathbf{x}}{\mathbf{y}}$$

Hence, we have that  $LSD((\mathcal{R}_\alpha(\vec{a}), \mathcal{R}_\alpha(\vec{b})), (\vec{u}, \vec{v}))$

$$= \left| \frac{R_\alpha(\mathbf{a})}{R_\alpha(\mathbf{b})} - \frac{\mathbf{u}}{\mathbf{v}} \right|_{\mathbb{C}} = \left| \frac{\mathbf{a}}{\mathbf{b}} - \frac{\mathbf{u}}{\mathbf{v}} \right|_{\mathbb{C}} = LSD((\vec{a}, \vec{b}), (\vec{u}, \vec{v}))$$

One can verify that the general LSD defined in (5) has the same invariance properties in any finite dimensional non-degenerate quadratic space. In order to examine how LSD can also be used for 3D gestures and for the symetrized version, we introduce the following definitions.

#### G. Discrete Curves

A continuous *curve* is a continuous function  $\vec{x}: I \rightarrow \mathbb{R}^n$  from a interval  $I$  to the  $d$ -dimensional Euclidean space  $\mathbb{R}^d$ . A *discrete curve* is a string (i.e. a sequence)  $\vec{X} = \{\vec{x}_0, \dots, \vec{x}_{L-1}\}$  of  $L$  ordered points, Where  $L$  is called the *length* of the string  $\vec{X}$ , in the  $d$ -dimensional Euclidean space  $\mathbb{R}^d$  laying on some curve. A *plane discrete curve* is a discrete curve whose points lay in the Euclidean plane  $\mathbb{R}^2$ . Using Cartesian coordinates, a plane discrete curve  $\vec{X} = \{\vec{x}_0, \dots, \vec{x}_{L-1}\}$  will appear as a sequence of couples of real numbers:

$$\vec{X} = \{\vec{x}_0, \dots, \vec{x}_{L-1}\} = \{(x_{1,0}, x_{2,0}), \dots, (x_{1,L-1}, x_{2,L-1})\}$$

Two discrete curves  $\vec{X} = \{\vec{x}_0, \dots, \vec{x}_{L-1}\}$ ,  $\vec{Y} = \{\vec{y}_0, \dots, \vec{y}_{M-1}\}$  are *isoparametrized* iff their strings have the same length, i.e.  $L=M$ . A plane discrete curve  $\vec{X} = \{\vec{x}_0, \dots, \vec{x}_{L-1}\}$  is *regular* if  $\vec{x}_{i+1} - \vec{x}_i \neq \vec{0} = (0, 0), \forall i=0, \dots, L-2$ .

**The oriented shape distance between two discrete curves.** Given two regular isoparametrized plane discrete curves  $\vec{X} = \{\vec{x}_0, \dots, \vec{x}_{L-1}\}$ ,  $\vec{Y} = \{\vec{y}_0, \dots, \vec{y}_{L-1}\}$ , the function  $d^+(\vec{X}, \vec{Y}) =$

$$\sum_{0 \leq i \leq L-3} LSD((\vec{x}_{i+1}-\vec{x}_i, \vec{x}_{i+2}-\vec{x}_{i+1}), (\vec{y}_{i+1}-\vec{y}_i, \vec{y}_{i+2}-\vec{y}_{i+1}))$$

is called the (positively oriented) *Shape Distance* between  $\vec{X}$  and  $\vec{Y}$ . If  $\vec{X}$  and  $\vec{Y}$  are both basic gestures laying on a same plane, then  $d(\vec{X}, \vec{Y}) = LSD((\vec{x}_1-\vec{x}_0, \vec{x}_2-\vec{x}_1), (\vec{y}_1-\vec{y}_0, \vec{y}_2-\vec{y}_1))$ .

**The unoriented shape distance between two discrete curves.** Since LSD is sensible to the orientation of the basic gestures,  $d^+$  is sensible to the orientation of the discrete curves. We therefore propose a shape distance that does not depend on the orientation on the two regular isoparametrized plane discrete curves. Given two regular isoparametrized plane discrete curves  $\vec{X} = \{\vec{x}_0, \dots, \vec{x}_{L-1}\}$ ,  $\vec{Y} = \{\vec{y}_0, \dots, \vec{y}_{L-1}\}$ , the function  $d^-(\vec{X}, \vec{Y}) = \sum_{0 \leq i \leq L-3} LSD((\vec{x}_{i+1}-\vec{x}_i, \vec{x}_{i+2}-\vec{x}_{i+1}), (\vec{y}_{L-(i+2)}-\vec{y}_{L-(i+1)}, \vec{y}_{L-(i+3)}-\vec{y}_{L-(i+2)}))$

is called the (negatively oriented) *Shape Distance* between  $\vec{X}$  and  $\vec{Y}$ , and  $d(\vec{X}, \vec{Y}) = \min \{d^+(\vec{X}, \vec{Y}), d^-(\vec{X}, \vec{Y})\}$  is called the (unoriented) *Shape Distance* between  $\vec{X}$  and  $\vec{Y}$ .

**The shape distance between two discrete curves (not necessarily isoparametrized).** Two plane gestures, corresponding to two regular plane discrete curves  $\vec{X}$  and  $\vec{Y}$ ,  $\vec{X} = \{\vec{x}_0, \dots, \vec{x}_{L-1}\}$ ,  $\vec{Y} = \{\vec{y}_0, \dots, \vec{y}_{M-1}\}$  are often not isoparametrized ( $L \neq M$ ), especially when coming from different gesture sets. Here we propose a “linear” way to compare them through the above shape distance. If  $L \neq M$ , we can consider the points of each discrete curve  $\vec{X}$  and  $\vec{Y}$  as the vertices of polygonal (continuous) curves  $\vec{x}$  and  $\vec{y}$ , respectively. We can fix whatever  $P \in \mathbb{N}$  (with  $P \geq 3$ ), interpolate  $P$  points on each polygonal continuous curve  $\vec{x}$  and  $\vec{y}$ , and thus produce two regular isoparametrized plane discrete curve of the same length  $P$ :  $\vec{X}^{(P)} = \{\vec{x}_0^{(P)}, \dots, \vec{x}_{P-1}^{(P)}\}$ ,  $\vec{Y}^{(P)} = \{\vec{y}_0^{(P)}, \dots, \vec{y}_{P-1}^{(P)}\}$ . Then we can calculate the distance depending on the new common length  $P$ :  $d_P(\vec{X}, \vec{Y}) = d(\vec{X}^{(P)}, \vec{Y}^{(P)})$ . There are different criteria to interpolate points on a continuous curve; here, we consider the linear criterion.

**Definition of string  $\vec{X}^{(P)}$  from the given string  $\vec{X}$ .** Let  $\vec{X}$  be a regular plane discrete curve  $\vec{X} = \{\vec{x}_0, \dots, \vec{x}_{L-1}\}$  with  $L \geq 2$ . Let  $P \in \mathbb{N}$  (with  $P \geq 3$ ), we define  $\forall j=0, \dots, P-1$ ,

$$\vec{x}_j^{(P)} = \frac{P-(j+1)}{P-1} \vec{x}_{\lfloor \frac{N-1}{P-1} j \rfloor} + \frac{j}{P-1} \vec{x}_{\lceil \frac{N-1}{P-1} j \rceil}$$

where  $\lfloor \cdot \rfloor$  is the floor function, and  $\lceil \cdot \rceil$  is the ceiling function.

#### IV. CONCLUSION AND FUTURE WORK

This paper explained how a Local Shape Distance (LSD and NLSD) could compute the dissimilarity between gestures represented as  $n$ -dimensional vectors with several properties: point-number and stroke-number invariance, stroke-order and stroke-direction invariance, position, scale, and rotation invariance. An instantiation of LSD (and NLSD) to  $n=2$  gave rise to a 2D stroke gesture recognizer. This work will benefit practitioners by providing a new gesture recognizer satisfying

That is, have both length  $L = 3$ , the basic plane gesture corresponding to the plane discrete curve  $\vec{X}$  being  $(\vec{x}_1-\vec{x}_0, \vec{x}_2-\vec{x}_1)$ , and the basic plane gesture associated to plane discrete curve  $\vec{Y}$  being  $(\vec{y}_1-\vec{y}_0, \vec{y}_2-\vec{y}_1)$ .

Other ways could use non-linear splines to continuously connect the ordered points of a discrete curve.

the requirements in a comparable way while preserving the invariance properties. If for any reason a gesture should become for instance scale variant, the constraint can be imposed on the vectors without any loss of generality and without complexifying the recognizer with additional computations.

#### REFERENCES

- [1] L. Anthony and J. O. Wobbrock. A lightweight multistroke recognizer for user interface prototypes. In *Proc. of GI '10*, pages 245–252, 2010.
- [2] C. Appert and O. Bau. Scale detection for a priori gesture recognition. In *Proc. of CHI '10*, pages 879–882. ACM, 2010.
- [3] F. Beuvs and J. Vanderdonck. Designing graphical user interfaces integrating gestures. In *Proc. of SIGDOC '12*, 2012.
- [4] A. Bragdon, R. Zeleznik, B. Williamson, T. Miller, and J. J. LaViola, Jr. Gesturebar: Improving the approachability of gesture-based interfaces. In *Proc. of CHI '09*, 2009.
- [5] S. h.-h. Chang, R. Blagojevic, and B. Plimmer. Rata.gesture: A gesture recognizer developed using data mining. *Artif. Intell. Eng. Des. Anal. Manuf.*, 26(3):351–366, Aug. 2012.
- [6] M. Cirelli and R. Nakamura. A survey on multi-touch gesture recognition and multi-touch frameworks. In *Proc. of ITS '14*.
- [7] A. Coyette, S. Schimke, J. Vanderdonck, and C. Vielhauer. *Trainable Sketch Recognizer for Graphical User Interface Design*.
- [8] L. Dorst, D. Fontijne, and S. Mann. *Geometric Algebra for Computer Science: An Object-Oriented Approach to Geometry*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1st edition, 2007.
- [9] V. Fuccella, M. De Rosa, and G. Costagliola. Novice and expert performance of keystretch: A gesture-based text entry method for touch-screens. *IEEE Trans. on HMI*, 44(4):511–523, Aug 2014.
- [10] J. Herold and T. F. Stahovich. The 1&cent; recognizer: A fast, accurate, and easy-to-implement handwritten gesture recognition technique. In *Proc. of SBIM '12*, pages 39–46, 2012.
- [11] H. Hse, M. Shilman, and A. R. Newton. Robust sketched symbol fragmentation using templates. In *Proc. of IUI '04*.
- [12] K. Kim and H.-I. Choi. Online hand gesture recognition using enhanced &#36;n recogniser based on a depth camera. *Int. J. Comput. Vision Robot.*, 6(3):214–222, Jan. 2016.
- [13] S. Kratz and M. Rohs. Protractor3d: A closed-form solution to rotation-invariant 3d gestures. In *Proc. of IUI '11*.
- [14] L. A. Leiva, D. Martín-Albo, and R. Plamondon. Gestures À go go: Authoring synthetic human-like stroke gestures using the kinematic theory of rapid movements. *ACM TIS*.
- [15] J. A. Lester. Triangles i: Shapes. *aequationes mathematicae*, 52(1):30–54, Feb 1996.
- [16] C. Pittman, E. M. Taranta II, and J. J. LaViola, Jr. A \$-family friendly approach to prototype selection. In *Proc. of IUI '16*.
- [17] D. Rubine. Specifying gestures by example. In *Proc. of SIGGRAPH '91*, pages 329–337, New York, NY, USA, 1991. ACM.
- [18] B. Signer, U. Kurmann, and M. Norrie. igesture: A general gesture recognition framework. In *Proc. of ICDAR '07*, pages 954–958, 2007.
- [19] E. Taranta and J. LaViola. Penny pincher: a blazing fast, highly accurate \$-family recognizer. In *Proc. of GI 2015*.
- [20] E. M. Taranta, A. N. Vargas, and J. J. LaViola. Streamlined and accurate gesture recognition with penny pincher. *Computers Graphics*, 55:130 – 142, 2016.
- [21] H. Tu, X. Ren, and S. Zhai. Differences and similarities between finger and pen stroke gestures on stationary and mobile devices. *ACM Trans. Comput.-Hum. Interact.*, 22(5):1–39, Aug. 2015.
- [22] J. Vanderdonck, P. Roselli, and J. L. Pérez-Medina. !FTL, an articulation-invariant stroke gesture recognizer with controllable position, scale, and rotation invariances. In *Proc. of ICMI '18*.
- [23] R.-D. Vatavu. The effect of sampling rate on the performance of template-based gesture recognizers. In *Proc. of ICMI '11*.
- [24] R.-D. Vatavu, L. Anthony, and J. Wobbrock. \$Q: A super-quick, articulation-invariant stroke-gesture recognizer for low-resource devices. In *Proc. of MobileHCI '18*, pages 623–635. ACM, 2018.
- [25] R.-D. Vatavu, L. Anthony, and J. O. Wobbrock. Gestures as point clouds: A \$P recognizer for user interface prototypes. In *Proc. of ICMI '12*, pages 273–280, New York, NY, USA, 2012. ACM.
- [26] R.-D. Vatavu, G. Casiez, and L. Grisoni. Small, medium, or large?: Estimating the user-perceived scale of stroke gestures. In *Proc. of CHI '13*, pages 277–280, New York, NY, USA, 2013. ACM.
- [27] J. O. Wobbrock, A. D. Wilson, and Y. Li. Gestures without libraries, toolkits or training: A \$1 recognizer for user interface prototypes. In *Proc. of UIST '07*, pages 159–168, New York, 2007. ACM.
- [28] S. Zhai, P. O. Kristensson, C. Appert, T. H. Anderson, and X. Cao. Foundational issues in touch-surface stroke gesture design: an integrative review. *Foundations and Trends in HCI*, 5(2):97–205, 2012.