Composed Computational Verb Similarities

Tao Yang

***Abstract*— Computational verb similarities are used to measure the degree of similarities between two waveforms. Since the similarities of waveforms might be measured based on their distance, trends, frequencies and many other factors, it is easy to consider the contributions of these factors to computational verb similarities separately and then combine them into composed computational verb similarities. In this paper, different ways of calculating composed computational verb similarities are presented.** *Copyright © 2009 Yang's Scientific Research Institute, LLC. All rights reserved.*

***Index Terms*— Computational verb, computational verb similarity.**

I. INTRODUCTION

COMPUTATIONAL verb similarities(CVS's) play very important roles in computational verb reasoning, computational verb clustering and computational verb knowledge representation. Although there were many ways of defining CVS's, so far, there is no CVS that can be used to fit universally well into all applications. To design a good CVS, the following factors must be taken into account.

1) Local and global trends, and derivatives of waveforms.
2) Local and global shapes of waveforms.
3) Frequency and spectrum of waveforms.
4) Amplitudes, range, minimum, maximum, average value, and other statistical measurements of waveforms.

It is difficult to design a verb similarity which can comprehensively consider all aspects listed in above. In this paper, I will provide a divide-and-conquer method of making composed computational verb similarities based on the considerations of these factors individually.

The organization of this paper is as follows. In Section II, the brief history of computational verb theory will be given. In Section III, the principles of calculating composed computational verb similarities will be presented. In Section IV, composed computational verbs for two-sampled computational verbs will be presented. In Section V, composed computational verbs for three-sampled computational verbs will be presented. In Section VI, some concluding remarks will be included.

II. A BRIEF HISTORY OF COMPUTATIONAL VERB THEORY

As the first paradigm shift for solving engineering problems by using verbs, the computational verb theory[31] and physical linguistics[34], [51], [25] have undergone a rapid growth since the birth of computational verb in the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley in 1997[16], [17]. The paradigm of implementing verbs in machines was coined as *computational verb theory*[31]. The building blocks of computational theory are *computational verbs*[26], [20], [18], [27], [32]. The relation between verbs and adverbs was mathematically defined in [19]. The logic operations between verb statements were studied in [21]. The applications of verb logic to verb reasoning were addressed in [22] and further studied in [31]. A logic paradox was solved based on verb logic[28]. The mathematical concept of set was generalized into verb set in[24]. Similarly, for measurable attributes, the number systems can be generalized into verb numbers[29]. The applications of computational verbs to predictions were studied in [23]. In [33] fuzzy dynamic systems were used to model a special kind of computational verb that evolves in fuzzy spaces. The relation between computational verb theory and traditional linguistics was studied in [31], [34]. The theoretical basis of developing computational cognition from a unified theory of fuzzy and computational verb theory is the theory of the UNICOGSE that was studied in [34], [39]. The issues of simulating cognition using computational verbs were studied in [35]. In [62] the correlation between computational verbs was studied. A method of implementing feelings in machines was proposed based on grounded computational verbs and computational nouns in [41]. In [48] a theory of how to design stable computational verb controllers was given. In [42] the rule-wise linear computational verb systems and their applications to the design of stable computational verb controllers and chaos in computational verb systems were presented. In [46] the concept of computational verb entropy was used to construct computational verb decision tree for data-mining applications. In [45] the relation between computational verbs and fuzzy sets was studied by using computational verb collapses and computational verb extension principles. In [47] the distances and similarities of saturated computational verbs were defined as normalized measures of the distances and similarities between computational verbs. Based on saturated computational verbs, the verb distances and similarities are related to each other with a simple relation. The distances and similarities between verbs with different life spans can be defined based on saturated computational verbs as well. In [49] the methods of using computational verbs to cluster trajectories and curves were presented. To cluster a bank of trajectories into a few representative computational

---

Manuscript received March 20, 2009; revised April 15, 2009, May 09, 2009.

Tao Yang, Department of Electronic Engineering, Xiamen University, Xiamen 361005, P.R. China. Department of Cognitive Economics, Department of Electrical Engineering and Computer Sciences, Yang's Scientific Research Institute, 1303 East University Blvd., #20882, Tucson, Arizona 85719-0521, USA. Email: taoyang@xmu.edu.cn,taoyang@yangsky.com,taoyang@yangsky.us.

Publisher Item Identifier S 1542-5908(09)10112-4/$20.00

verbs is to discover knowledge from database of time series. We use cluster centers to represent complex waveforms at symbolic levels. In [14] computational verb controllers were used to control a chaotic circuit model known as Chua's circuit. Computational verb controllers were designed based on verb control rules for different dynamics of the region-wise linear model of the control plant. In [13] computational verb controllers were used to synchronize discrete-time chaotic systems known as Hénon maps. Different verb control rules are designed for synchronizing different kinds of dynamics. In [53], how can computational verb theory functions as the most essential building block of cognitive engineering and cognitive industries was addressed. Computational verb theory will play a critical important role in personalizing services in the next fifty years. In [50], [52] computational verb theory was used to design an accurate flame-detecting systems based on CCTV signal. In [56] the learning algorithms were presented for learning computational verb rules from training data. In [54] the structures and learning algorithms of computational verb neural networks were presented. In [61] the ambiguities of the states and dynamics of computational verbs were studied. In [55] the history and milestones in the first ten years of the studies of computational verb theory were given. In [4] a case study of modeling adverbs as modifiers of computational verbs was presented. In [15] computational verb rules were used to improve the training processes of neural networks. In [57] the classifications and interactions between computational verb rule bases were presented. In [58] the simplest verb rules and their verb reasoning were connected to many intuitive applications of verb rules before the invention of computational verbs. In [59] the interactions between computational verbs were used as a powerful tools to understand the merging and splitting effects of verbs. In [3] computational verb rules were trained by using prescribed training samples of functions. In [60] the trend-based computational verb similarity was given as a way to decrease the computational complexities of verb similarities. In [5] computational verb PID controller was used to control linear motors. In [12] computational verb controller was used to control an auto-focusing system.

The theory of computational verb has been taught in some university classrooms since 2005[1]. The latest active applications of computational verb theory are listed as follows.

1) Computational Verb Controllers. The applications of computational verbs to different kinds of control problems were studied on different occassions[30], [31]. For the advanced applications of computational verbs to control problems, a few papers reporting the latest advances had been published[37], [36], [48], [42], [63]. The design of computational verb controllers was also presented in a textbook in 2005[1].
2) Computational Verb Image Processing and Image Understanding. The recent results of image processing by using computational verbs can be found in[38]. The applications of computational verbs to image understanding can be found in [40]. The authors of [2] applied computational verb image processing to design the vision systems of RoboCup small-size robots.
3) Stock Market Modeling and Prediction based on computational verbs. The product of Cognitive Stock Charts[8] was based on the advanced modeling and computing reported in [43]. Computational verb theory was used to study the trends of stock markets known as Russell reconstruction patterns [44].

Computational verb theory has been successfully applied to many industrial and commercial products. Some of these products are listed as follows.

1) Visual Card Counters. The *YangSky-MAGIC* card counter[10], developed by Yang's Scientific Research Institute and Wuxi Xingcard Technology Co. Ltd., was the first visual card counter to use computational verb image processing technology to achieve high accuracy of card and paper board counting based on cheap webcams.
2) CCTV Automatic Driver Qualify Test System. The *DriveQfy* CCTV automatic driver qualify test system[11] was the first vehicle trajectory reconstruction and stop time measuring system using computational verb image processing technology.
3) Visual Flame Detecting System. The *FireEye* visual flame detecting system[6] was the first CCTV or webcam based flame detecting system, which works under color and black & white conditions, for surveillance and security monitoring system.
4) Smart Pornographic Image and Video Detection Systems. The *PornSeer*[9] pornographic image and video detection systems are the first cognitive feature based smart porno detection and removal software.
5) Webcam Barcode Scanner. The *BarSeer*[7] webcam barcode scanner took advantage of the computational verb image processing to make the scan of barcode by using cheap webcam possible.
6) Cognitive Stock Charts. By applying computational verbs to the modeling of trends and cognitive behaviors of stock trading activities, cognitive stock charts can provide the traders with the "feelings" of stock markets by using simple and intuitive indexes.
7) TrafGo ITS SDK. Computational verbs were applied to model vehicle trajectories and dynamics of optical field and many other aspects of dynamics in complex environments for applications in intelligent transportation systems (ITS).

III. PRINCIPLES OF CALCULATING COMPOSED COMPUTATIONAL VERB SIMILARITIES

Given two computational verbs V_1 and V_2, then the CVS between V_1 and V_2 is calculated based on the combination

---

[1] Some computational verb theory related college courses are

- Dr. G. R. Chen, EE 64152 - Introduction to Fuzzy Informatics and Intelligent Systems, Department of Electronic Engineering, City University of Hong Kong.
- Dr. D. H. Guo, Artificial Intelligence, Department of Electronic Engineering, Xiamen University.
- Prof. T. Yang, Computational Methodologies in Intelligent Systems, Department of Electronic Engineering, Xiamen University.
- Dr. Mahir Sabra, EELE 6306: Intelligent Control, Electrical and Computer Engineering Department, The Islamic University of Gaza.

of similarities between different properties of computational verbs as follow.

$$S(\mathsf{V}_1,\mathsf{V}_2)=\sigma(\gamma_d(\mathsf{V}_1,\mathsf{V}_2),\gamma_t(\mathsf{V}_1,\mathsf{V}_2),\gamma_f(\mathsf{V}_1,\mathsf{V}_2),\dots) \quad (1)$$

where $\sigma(\cdots)$ is a function to compose different aspects of measuring CVS's such as

- $\gamma_d(\cdot,\cdot)$ to measure CVS in the term of distance between two computational verbs;
- $\gamma_t(\cdot,\cdot)$ to measure CVS in the term of comparing trends of two computational verbs;
- $\gamma_f(\cdot,\cdot)$ to measure CVS in the term of comparing frequencies of two computational verbs;
- others.

Let $\gamma_\checkmark(\cdot,\cdot)$ denotes any of the γ-functions in Eq. (1) and assume that $\gamma_\checkmark(\cdot,\cdot)$ satisfies the following conditions

- $\gamma_\checkmark(\mathsf{V}_1,\mathsf{V}_2)\in[0,1]$,
- $\gamma_\checkmark(\mathsf{V}_1,\mathsf{V}_2)=1$ if and only if V_1 and V_2 are the same,
- $\gamma_\checkmark(\mathsf{V}_1,\mathsf{V}_2)=0$ if and only if V_1 and V_2 are mostly different,

then the function $\sigma(\cdots)$ is chosen as the following t-norm function

$$\begin{aligned} S(\mathsf{V}_1,\mathsf{V}_2) &= \gamma_d(\mathsf{V}_1,\mathsf{V}_2)\wedge\gamma_t(\mathsf{V}_1,\mathsf{V}_2)\wedge\gamma_f(\mathsf{V}_1,\mathsf{V}_2)\wedge\dots\\ &\triangleq \bigwedge_{\checkmark}\gamma_\checkmark(\mathsf{V}_1,\mathsf{V}_2). \end{aligned} \quad (2)$$

Observe that different kinds of CVS's might be defined based on different choices of t-norms and γ-functions.

A. Distances

Let $d(\mathsf{V}_1,\mathsf{V}_2)$ be a measure proportional to the distance between V_1 and V_2, then $\gamma_d(\mathsf{V}_1,\mathsf{V}_2)$ satisfies the following conditions.

- If $d(\mathsf{V}_1,\mathsf{V}_2)=0$, then $\gamma_d(\mathsf{V}_1,\mathsf{V}_2)=1$.
- If $d(\mathsf{V}_1,\mathsf{V}_2)=\infty$, then $\gamma_d(\mathsf{V}_1,\mathsf{V}_2)=0$.
- $\gamma_d(\mathsf{V}_1,\mathsf{V}_2)$ is monotonically decreasing with respect to $d(\mathsf{V}_1,\mathsf{V}_2)$.

Some examples of $\gamma_d(\mathsf{V}_1,\mathsf{V}_2)$ are listed as follows.

$$\begin{aligned} \gamma_d(\mathsf{V}_1,\mathsf{V}_2) &= \frac{2}{1+e^{d(\mathsf{V}_1,\mathsf{V}_2)}}\\ \gamma_d(\mathsf{V}_1,\mathsf{V}_2) &= \frac{1}{1+[d(\mathsf{V}_1,\mathsf{V}_2)]^p}, p\in\mathbb{N}, p\ge 1,\\ \gamma_d(\mathsf{V}_1,\mathsf{V}_2) &= \max(1-[d(\mathsf{V}_1,\mathsf{V}_2)]^p,0),\\ \gamma_d(\mathsf{V}_1,\mathsf{V}_2) &= e^{-d(\mathsf{V}_1,\mathsf{V}_2)}. \end{aligned} \quad (3)$$

$d(\mathsf{V}_1,\mathsf{V}_2)$ is a function that might have the following forms.

$$\begin{aligned} d(\mathsf{V}_1,\mathsf{V}_2) &= |\mathcal{E}_1(t_0)-\mathcal{E}_2(t_0)|, t_0\in[0,T],\\ d(\mathsf{V}_1,\mathsf{V}_2) &= \frac{1}{T}\left|\int_{t=0}^{T}\mathcal{E}_1(t)dt-\int_{t=0}^{T}\mathcal{E}_2(t)dt\right|,\\ d(\mathsf{V}_1,\mathsf{V}_2) &= \int_{t=0}^{t}|\mathcal{E}_1(t)-\mathcal{E}_2(t)|dt,\\ d(\mathsf{V}_1,\mathsf{V}_2) &= \int_{t=0}^{t}[\mathcal{E}_1(t)-\mathcal{E}_2(t)]^2dt,\\ d(\mathsf{V}_1,\mathsf{V}_2) &= \left(\int_{t=0}^{t}|\mathcal{E}_1(t)-\mathcal{E}_2(t)|^pdt\right)^{1/p}, p\ge 1. \end{aligned} \quad (4)$$

B. Trends

The contribution to similarity from the trends might be calculated based on the distance of the derivatives of the evolving functions as follow.

$$d_t(\mathsf{V}_1,\mathsf{V}_2) = d\left(\frac{d\mathcal{E}_1(t)}{dt},\frac{d\mathcal{E}_2(t)}{dt}\right) \quad (5)$$

where the function $d(\cdot,\cdot)$ is the same as those defined in Eq. (4); namely,

$$\begin{aligned} d_t(\mathsf{V}_1,\mathsf{V}_2) &= |\dot{\mathcal{E}}_1(t_0)-\dot{\mathcal{E}}_2(t_0)|, t_0\in[0,T],\\ d_t(\mathsf{V}_1,\mathsf{V}_2) &= \frac{1}{T}\left|\int_{t=0}^{T}\dot{\mathcal{E}}_1(t)dt-\int_{t=0}^{T}\dot{\mathcal{E}}_2(t)dt\right|,\\ d_t(\mathsf{V}_1,\mathsf{V}_2) &= \int_{t=0}^{t}|\dot{\mathcal{E}}_1(t)-\dot{\mathcal{E}}_2(t)|dt,\\ d_t(\mathsf{V}_1,\mathsf{V}_2) &= \int_{t=0}^{t}[\dot{\mathcal{E}}_1(t)-\dot{\mathcal{E}}_2(t)]^2dt,\\ d_t(\mathsf{V}_1,\mathsf{V}_2) &= \left(\int_{t=0}^{t}|\dot{\mathcal{E}}_1(t)-\dot{\mathcal{E}}_2(t)|^pdt\right)^{1/p}, p\ge 1. \end{aligned} \quad (6)$$

Based on $d_t(\mathsf{V}_1,\mathsf{V}_2)$, $\gamma_t(\mathsf{V}_1,\mathsf{V}_2)$ must satisfy the following conditions.

- If $d_t(\mathsf{V}_1,\mathsf{V}_2)=0$, then $\gamma_t(\mathsf{V}_1,\mathsf{V}_2)=1$.
- If $d_t(\mathsf{V}_1,\mathsf{V}_2)=\infty$, then $\gamma_t(\mathsf{V}_1,\mathsf{V}_2)=0$.
- $\gamma_t(\mathsf{V}_1,\mathsf{V}_2)$ is monotonically decreasing with respect to $d_t(\mathsf{V}_1,\mathsf{V}_2)$.

Some examples of $\gamma_t(\mathsf{V}_1,\mathsf{V}_2)$ are listed as follows.

$$\begin{aligned} \gamma_t(\mathsf{V}_1,\mathsf{V}_2) &= \frac{2}{1+e^{d_t(\mathsf{V}_1,\mathsf{V}_2)}}\\ \gamma_t(\mathsf{V}_1,\mathsf{V}_2) &= \frac{1}{1+[d_t(\mathsf{V}_1,\mathsf{V}_2)]^p}, p\in\mathbb{N}, p\ge 1,\\ \gamma_t(\mathsf{V}_1,\mathsf{V}_2) &= \max(1-[d_t(\mathsf{V}_1,\mathsf{V}_2)]^p,0),\\ \gamma_t(\mathsf{V}_1,\mathsf{V}_2) &= e^{-d_t(\mathsf{V}_1,\mathsf{V}_2)}. \end{aligned} \quad (7)$$

C. Frequencies

The contribution to similarity from the frequency domain might be calculated based on the distance of the Fourier transforms of the evolving functions as follow.

$$d_f(\mathsf{V}_1,\mathsf{V}_2) = d_{ft}(E_1,E_2) \quad (8)$$

where E_1 and E_2 are Fourier transforms of $\mathcal{E}_1$ and $\mathcal{E}_2$, respectively. $d_{ft}(\cdot,\cdot)$ is a function used to measure the distance between two Fourier transforms. We assume that $\mathcal{E}_1$ and $\mathcal{E}_2$ satisfy

$$\int_{-\infty}^{\infty}|\mathcal{E}_1(t)|dt<\infty,\ \int_{-\infty}^{\infty}|\mathcal{E}_2(t)|dt<\infty \quad (9)$$

then

$$\begin{aligned} E_1(\jmath\omega) &= \int_{-\infty}^{\infty}\mathcal{E}_1(t)e^{-\jmath\omega t}dt,\\ E_2(\jmath\omega) &= \int_{-\infty}^{\infty}\mathcal{E}_2(t)e^{-\jmath\omega t}dt. \end{aligned} \quad (10)$$

For example, $d_{ft}(\cdot,\cdot)$ might be

$$d_{ft}(E_1,E_2) = \int_{-\infty}^{\infty}[E_1(\jmath\omega)-E_2(\jmath\omega)]\overline{E_1(\jmath\omega)-E_2(\jmath\omega)}d\omega. \quad (11)$$

Based on $d_f(\mathsf{V}_1,\mathsf{V}_2)$, $\gamma_f(\mathsf{V}_1,\mathsf{V}_2)$ must satisfy the following conditions.

- If $d_f(\mathsf{V}_1,\mathsf{V}_2)=0$, then $\gamma_f(\mathsf{V}_1,\mathsf{V}_2)=1$.
- If $d_f(\mathsf{V}_1,\mathsf{V}_2)=\infty$, then $\gamma_f(\mathsf{V}_1,\mathsf{V}_2)=0$.
- $\gamma_f(\mathsf{V}_1,\mathsf{V}_2)$ is monotonically decreasing with respect to $d_f(\mathsf{V}_1,\mathsf{V}_2)$.

Some examples of $\gamma_f(\mathsf{V}_1,\mathsf{V}_2)$ are listed as follows.

$$\begin{aligned}
\gamma_f(\mathsf{V}_1,\mathsf{V}_2) &= \frac{2}{1+e^{d_f(\mathsf{V}_1,\mathsf{V}_2)}}\\
\gamma_f(\mathsf{V}_1,\mathsf{V}_2) &= \frac{1}{1+[d_f(\mathsf{V}_1,\mathsf{V}_2)]^p},\\
\gamma_f(\mathsf{V}_1,\mathsf{V}_2) &= \max(1-[d_f(\mathsf{V}_1,\mathsf{V}_2)]^p,0),\\
\gamma_f(\mathsf{V}_1,\mathsf{V}_2) &= e^{-d_f(\mathsf{V}_1,\mathsf{V}_2)}, p\in\mathbb{N}, p\geq 1. \quad (12)
\end{aligned}$$

D. Discrete-time Cases

Let V_1 and V_2 are two n-sampled discrete-time evolving sequence represented as

$$\mathsf{V}_1 \triangleq \{x_1(k)\}_{k=1}^n, \mathsf{V}_2 \triangleq \{x_2(k)\}_{k=1}^n \quad (13)$$

where $x_1(k)$ and $x_2(k)$, $k=1,\ldots,n$, denote the kth samples of the evolving functions of V_1 and V_2, respectively.

1) Distances: The discrete-time cases corresponding to those listed in Eq. (4) are given as follows.

$$\begin{aligned}
d(\mathsf{V}_1,\mathsf{V}_2) &= |x_1(k)-x_2(k)|, k\in\{1,\ldots,n\},\\
d(\mathsf{V}_1,\mathsf{V}_2) &= \frac{1}{n}\left|\sum_{k=1}^n x_1(k)-\sum_{k=1}^n x_2(k)\right|,\\
d(\mathsf{V}_1,\mathsf{V}_2) &= \sum_{k=1}^n |x_1(k)-x_2(k)|,\\
d(\mathsf{V}_1,\mathsf{V}_2) &= \sum_{k=1}^n [x_1(k)-x_2(k)]^2,\\
d(\mathsf{V}_1,\mathsf{V}_2) &= \left(\sum_{k=1}^n |x_1(k)-x_2(k)|^p\right)^{1/p}, p\geq 1.
\end{aligned} \quad (14)$$

For the discrete-time cases, we have an additional distance measurement. Assume that the evolving function of two computational verbs are two random vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ of the same distribution with the covariance matrix Γ, then the Mahalanobis distance is defined as

$$d(\boldsymbol{x},\boldsymbol{y}) = \sqrt{(\boldsymbol{x}-\boldsymbol{y})^\top\Gamma^{-1}(\boldsymbol{x}-\boldsymbol{y})}. \quad (15)$$

If the covariance matrix is diagonal, then the Mahalanobis distance is called the *normalized Euclidean distance* given by

$$d(\boldsymbol{x},\boldsymbol{y}) = \sqrt{\sum_{i=1}^n \frac{(x_i-y_i)^2}{\sigma_i^2}}, \quad (16)$$

where σ_i is the standard deviation of x_i.

2) Trends: We have the following examples of $\gamma_t(\mathsf{V}_1,\mathsf{V}_2)$ as

$$\begin{aligned}
d_t(\mathsf{V}_1,\mathsf{V}_2) &= \sum_{k=1}^n |x_1(k)-x_1(1)-x_2(k)+x_2(1)|,\\
d_t(\mathsf{V}_1,\mathsf{V}_2)&\\
&= \sum_{k=2}^n |x_1(k)-x_1(k-1)-x_2(k)+x_2(k-1)|,\\
d_t(\mathsf{V}_1,\mathsf{V}_2)&\\
&= \left(\sum_{k=1}^n |x_1(k)-x_1(1)-x_2(k)+x_2(1)|^p\right)^{1/p},\\
d_t(\mathsf{V}_1,\mathsf{V}_2)&\\
&= \left(\sum_{k=2}^n |x_1(k)-x_1(k-1)-x_2(k)+x_2(k-1)|^p\right)^{1/p},\\
p\geq 1.&
\end{aligned} \quad (17)$$

3) Frequencies: In discrete-time cases, we change the Fourier transforms in Section III-C into discrete-time Fourier transforms in order to calculate CVS's.

IV. COMPOSED COMPUTATIONAL VERB SIMILARITIES OF TWO-SAMPLED COMPUTATIONAL VERBS

When a computational verb consists of only two sample points, we call it a *two-sampled computational verb*. Let us assume two two-sampled computational verbs to be

$$\mathsf{V}_1 \triangleq \{x_1(1),x_1(2)\}, \mathsf{V}_2 \triangleq \{x_2(1),x_2(2)\}. \quad (18)$$

Then the γ-functions for calculating CVS can be explicitly listed as follows.

A. γ_d

One of the simplest choices of $d(\mathsf{V}_1,\mathsf{V}_2)$ in Eq. (3) is

$$d(\mathsf{V}_1,\mathsf{V}_2) = |x_1(1)-x_2(1)|. \quad (19)$$

In this case, Eq. (3) leads to the follows.

$$\begin{aligned}
\gamma_d(\mathsf{V}_1,\mathsf{V}_2) &= \frac{2}{1+e^{|x_1(1)-x_2(1)|}},\\
\gamma_d(\mathsf{V}_1,\mathsf{V}_2) &= \frac{1}{1+|x_1(1)-x_2(1)|^p},\\
\gamma_d(\mathsf{V}_1,\mathsf{V}_2) &= \max(1-|x_1(1)-x_2(1)|^p,0),\\
\gamma_d(\mathsf{V}_1,\mathsf{V}_2) &= e^{-|x_1(1)-x_2(1)|^p}, p\in\mathbb{N}, p\geq 1. \quad (20)
\end{aligned}$$

Furthermore, for γ_d in Eq. (3), we can choose different ways to reflect the distance between V_1 and V_2 as follows.

$$\begin{aligned}
d(\mathsf{V}_1,\mathsf{V}_2) &= \sqrt{(x_1(1)-x_2(1))^2+(x_1(2)-x_2(2))^2},\\
d(\mathsf{V}_1,\mathsf{V}_2) &= \left(\left|\frac{|x_1(1)+x_1(2)|}{2}-\frac{|x_2(1)+x_2(2)|}{2}\right|^p\right)^{1/p},\\
&\quad p\in\mathbb{N}, p\geq 1.
\end{aligned} \quad (21)$$

B. γ_t

The distance related to the trends between two two-sampled computational verbs is calculated as

$$d_t(\mathsf{V}_1, \mathsf{V}_2) = \left(|[x_1(2) - x_1(1)] - [x_2(2) - x_2(1)]|^p\right)^{1/p}. \tag{22}$$

And the function γ_t can be chosen as those in Eq. (7).

C. Exponential Forms

Assume that exponential forms are used to calculate distances and trends, then we have

$$\begin{aligned} \gamma_d(\mathsf{V}_1, \mathsf{V}_2) &= \frac{2}{1 + e^{\kappa_1 |x_1(1) - x_2(1)|}}, \\ \gamma_t(\mathsf{V}_1, \mathsf{V}_2) &= \frac{2}{1 + e^{\kappa_2 |(x_1(2) - x_1(1)) - (x_2(2) - x_2(1))|}} \end{aligned} \tag{23}$$

where $\kappa_1 > 0$ and $\kappa_2 > 0$ are two constants.

Example 1: Consider the following four two-sampled computational verbs

$$\mathsf{V}_1 = (0, 1), \mathsf{V}_2 = (1, 0), \mathsf{V}_3 = (0, 0), \mathsf{V}_4 = (1, 1). \tag{24}$$

Let the observing verb be

$$\mathsf{V}_x = (0.5, 0.5), \tag{25}$$

and choose $\kappa_1 = \kappa_2 = 1$, then the composed CVS's are given by

$$\begin{aligned} S_c(\mathsf{V}_1, \mathsf{V}_x) &= \frac{2}{1 + e^{0.5}} \frac{2}{1 + e^1} = 0.4061, \\ S_c(\mathsf{V}_2, \mathsf{V}_x) &= \frac{2}{1 + e^{0.5}} \frac{2}{1 + e^1} = 0.4061, \\ S_c(\mathsf{V}_3, \mathsf{V}_x) &= \frac{2}{1 + e^{0.5}} \frac{2}{1 + e^0} = 0.7551, \\ S_c(\mathsf{V}_4, \mathsf{V}_x) &= \frac{2}{1 + e^{0.5}} \frac{2}{1 + e^0} = 0.7551. \end{aligned} \tag{26}$$

In this case, we put the same weight to the contributions of trends and distances to CVS.

D. Bell-shaped Functions

Assume that bell-shaped functions are used to calculate distances and trends, then we have

$$\begin{aligned} \gamma_d(\mathsf{V}_1, \mathsf{V}_2) &= \frac{1}{1 + \kappa_1 [x_1(1) - x_2(1)]^2}, \\ \gamma_t(\mathsf{V}_1, \mathsf{V}_2) &= \frac{1}{1 + \kappa_2 [(x_1(2) - x_1(1)) - (x_2(2) - x_2(1))]^2}. \end{aligned} \tag{27}$$

where $\kappa_1 > 0$ and $\kappa_2 > 0$ are two constants.

Example 2: Consider the following four two-sample computational verbs

$$\mathsf{V}_1 = (0, 1), \mathsf{V}_2 = (1, 0), \mathsf{V}_3 = (0, 0), \mathsf{V}_4 = (1, 1). \tag{28}$$

Let the observing verb be

$$\mathsf{V}_x = (0.5, 0.5), \tag{29}$$

and choose $\kappa_1 = \kappa_2 = 1$, then the composed CVS's are given by

$$\begin{aligned} S_c(\mathsf{V}_1, \mathsf{V}_x) &= \frac{1}{1 + 0.5^2} \frac{1}{1 + 1^2} = 0.4, \\ S_c(\mathsf{V}_2, \mathsf{V}_x) &= \frac{1}{1 + 0.5^2} \frac{1}{1 + 1^2} = 0.4, \\ S_c(\mathsf{V}_3, \mathsf{V}_x) &= \frac{1}{1 + 0.5^2} \frac{1}{1 + 0^2} = 0.8, \\ S_c(\mathsf{V}_4, \mathsf{V}_x) &= \frac{1}{1 + 0.5^2} \frac{1}{1 + 0^2} = 0.8. \end{aligned} \tag{30}$$

In this case, we put the same weight to the contributions of trends and distances to CVS.

V. COMPOSED COMPUTATIONAL VERB SIMILARITIES OF THREE-SAMPLED COMPUTATIONAL VERBS

Assume that a computational verb has three sampling points in its evolving function, then its evolving function has the following two forms

$$\mathsf{V}(t) = (x(1), x(1) + \Delta_0, x(1) + \Delta_1) \tag{31}$$

and

$$\mathsf{V}(t) = (x(1), x(1) + \Delta_0, x(1) + \Delta_0 + \Delta_1). \tag{32}$$

The parameters Δ_0 and Δ_1 in the three-sample evolving functions in Eqs. (31) and (32) reflects two ways of representing the trends. In Eq. (31) the trends are represented based on a common reference point, which is a global reference point. In Eq. (32) the trends are represented based on local reference points. There are many ways to calculate the CVS between two three-sampled computational verbs. In this section, only a few examples will be presented.

A. Case of Eq. (31)

Let the evolving functions of two three-sampled computational verbs be

$$\begin{aligned} \mathsf{V}_1(t) &= (x_1(1), x_1(2), x_1(3)) \\ &= (x_1(1), x_1(1) + \Delta_{10}, x_1(1) + \Delta_{11}), \\ \mathsf{V}_2(t) &= (x_2(1), x_2(2), x_2(3)) \\ &= (x_2(1), x_2(1) + \Delta_{20}, x_2(1) + \Delta_{21}). \end{aligned} \tag{33}$$

1) γ_d: For γ_d in Eq. (3), we can choose different ways to reflect the distance between V_1 and V_2 as follows.

$$\begin{aligned} & d(\mathsf{V}_1, \mathsf{V}_2) = \\ & \sqrt{(x_1(1) - x_2(1))^2 + (x_1(2) - x_2(2))^2 + (x_1(3) - x_2(3))^2}, \\ & d(\mathsf{V}_1, \mathsf{V}_2) = \\ & \left(\left| \frac{|x_1(1) + x_1(2) + x_1(3)|}{3} - \frac{|x_2(1) + x_2(2) + x_2(3)|}{3} \right|^p \right)^{1/p}, \\ & p \in \mathbb{N}, p \geq 1. \end{aligned} \tag{34}$$

2) γ_t: The distance related to the trends between two three-sampled computational verbs is calculated as

$$\begin{aligned} d_t(\mathsf{V}_1, \mathsf{V}_2) &= \left(|\Delta_{10} - \Delta_{20}|^p + |\Delta_{11} - \Delta_{21}|^p\right)^{1/p}, \\ & \quad p \geq 1. \end{aligned} \tag{35}$$

And the γ_t can be calculated as those in Eq. (7).

B. Case of Eq. (32)

In this case, the evolving functions of two three-sampled computational verbs are

$$\begin{aligned}
\mathsf{V}_1(t) &= (x_1(1), x_1(2), x_1(3)) \\
&= (x_1(1), x_1(1) + \Delta_{10}, x_1(1) + \Delta_{10} + \Delta_{11}), \\
\mathsf{V}_2(t) &= (x_2(1), x_2(2), x_2(3)) \\
&= (x_2(1), x_2(1) + \Delta_{20}, x_2(1) + \Delta_{20} + \Delta_{21}) \quad (36)
\end{aligned}$$

The CVS has the same form as those presented in Sec. V-A. However, the Δ parameters have different reference points.

VI. CONCLUDING REMARKS

Since computational verb similarities play important roles in computational verb reasoning and knowledge representation, it is important to find efficient ways of calculating them. Here, the author only presented a few examples to show different ways of dividing the measurements of verb similarities into a few simple aspects and computing the contributions to verb similarities from these aspects. By putting different weights to mix these aspects of computational verb similarities, composed computational verb similarities provide a flexible way to reflect different aspects of computational verb similarities.

REFERENCES

[1] Guanrong Chen and Trung Tat Pham. *Introduction to Fuzzy Systems*. Chapman & Hall/CRC, November 2005. ISBN:1-58488-531-9.

[2] Wanmi Chen, Yanqin Wei, Minrui Fei, and Huosheng Hu. Applications of computational verbs to image processing of RoboCup small-size robots. In *Intelligent Control and Automation*, volume 344/2006 of *Lecture Notes in Control and Information Sciences*, pages 494–499. Springer, Berlin / Heidelberg, 2006.

[3] Y. Guo and T. Yang. Training computational verb neural networks with computational verb rule bases,. *International Journal of Computational Cognition*, 6(4):12–23, December 2008 [available online at $http://www.YangSky.us/ijcc/ijcc64.htm$, $http://www.YangSky.com/ijcc/ijcc64.htm$].

[4] Yi Guo. A study of adverbs as modifiers of computational verbs. *International Journal of Computational Cognition*, 6(1):31–35, March 2008 [available online at $http://www.YangSky.com/ijcc/ijcc61.htm$, $http://www.YangSky.us/ijcc/ijcc61.htm$].

[5] J. Li. Research and application of computational verb PID controller of linear motor. Master's thesis, Kunming University of Science and Technology, Feb. 2008 [available online at $http://www.YangSky.com/researches/computationalverbs/verbfuzctrl/vbPIDMotor.pdf$]. (in Chinese).

[6] Yang's Scientific Research Institute LLC. **FireEye** *Visual Flame Detecting Systems*. http://www.yangsky.us/products/flamesky/index.htm, http://www.yangsky.com/products/flamesky/index.htm, 2005.

[7] Yang's Scientific Research Institute LLC. **BarSeer** *Webcam Barcode Scanner*. http://www.yangsky.us/demos/barseer/barseer.htm, http://www.yangsky.com/demos/barseer/barseer.htm, 2006.

[8] Yang's Scientific Research Institute LLC. *Cognitive Stock Charts*. http://www.yangsky.us/products/stock/, http://www.yangsky.com/products/stock/, 2006.

[9] Yang's Scientific Research Institute LLC. **PornSeer** *Pornographic Image and Video Detection Systems*. http://www.yangsky.us/products/dshowseer/porndetection/PornSeePro.htm, http://www.yangsky.com/products/dshowseer/porndetection/PornSeePro.htm, 2006.

[10] Yang's Scientific Research Institute LLC. and Wuxi Xingcard Technology Ltd. **YangSky-MAGIC** *Visual Card Counters*. http://www.yangsky.us/products/cardsky/cardsky.htm, http://www.yangsky.com/products/cardsky/cardsky.htm, 2004.

[11] Yang's Scientific Research Institute LLC. and Chinese Traffic Management Research Institute of the Ministry of Public Security(TMRI-China). **DriveQfy** *Automatic CCTV Driver Qualify Testing Systems*. http://www.yangsky.us/products/driveqfy/driveqfy.htm, http://www.yangsky.com/products/driveqfy/driveqfy.htm, 2005.

[12] Weibin Tang, Yinghao Liao, Zhicong Chen, Lihuan Cai, Tao Yang, and Donghui Guo. Auto-focusing system for microscope based on computational verb controllers. In *Anti-counterfeiting, Security and Identification, 2008. ASID 2008. 2nd International Conference on*, pages 84–87, Guiyang, China, 20-23 Aug. 2008. IEEE, IEEE Press.

[13] R. Tonelli and T. Yang. Synchronizing Hénon maps using computational verb controllers. *Phys. Rev. E.*, 2007. submitted.

[14] R. Tonelli and T. Yang. Controlling Chua's circuits using computational verb controllers. *International Journal of Robust and Nonlinear Control*, 18(17):1622–1636, Nov. 25 2008.

[15] H.-B. Wang and T. Yang. Training neural networks using computational verb rules. *International Journal of Computational Cognition*, 6(2):17–32, June 2008 [available online at $http://www.YangSky.us/ijcc/ijcc62.htm$, $http://www.YangSky.com/ijcc/ijcc62.htm$].

[16] T. Yang. Verbal paradigms—Part I: Modeling with verbs. Technical Report Memorandum No. UCB/ERL M97/64, Electronics Research Laboratory, College of Engineering, University of California, Berkeley, CA 94720, 9 Sept. 1997. page 1-15.

[17] T. Yang. Verbal paradigms—Part II: Computing with verbs. Technical Report Memorandum No. UCB/ERL M97/66, Electronics Research Laboratory, College of Engineering, University of California, Berkeley, CA 94720, 18 Sept. 1997. page 1-27.

[18] T. Yang. Computational verb systems: Computing with verbs and applications. *International Journal of General Systems*, 28(1):1–36, 1999.

[19] T. Yang. Computational verb systems: Adverbs and adverbials as modifiers of verbs. *Information Sciences*, 121(1-2):39–60, Dec. 1999.

[20] T. Yang. Computational verb systems: Modeling with verbs and applications. *Information Sciences*, 117(3-4):147–175, Aug. 1999.

[21] T. Yang. Computational verb systems: Verb logic. *International Journal of Intelligent Systems*, 14(11):1071–1087, Nov. 1999.

[22] T. Yang. Computational verb systems: A new paradigm for artificial intelligence. *Information Sciences—An International Journal*, 124(1-4):103–123, 2000.

[23] T. Yang. Computational verb systems: Verb predictions and their applications. *International Journal of Intelligent Systems*, 15(11):1087–1102, Nov. 2000.

[24] T. Yang. Computational verb systems: Verb sets. *International Journal of General Systems*, 20(6):941–964, 2000.

[25] T. Yang. Computational verb systems: Towards a unified paradigm for artificial languages. In *Proceedings of the Fifth Joint Conference on Information Sciences(JCIS 2000)*, pages 29–32, Atlantic City, NJ, Feb. 27-Mar. 03 2000.

[26] T. Yang. *Advances in Computational Verb Systems*. Nova Science Publishers, Inc., Huntington, NY, May 2001. ISBN 1-56072-971-6.

[27] T. Yang. Computational verb systems: Computing with perceptions of dynamics. *Information Sciences*, 134(1-4):167–248, Jun. 2001.

[28] T. Yang. Computational verb systems: The paradox of the liar. *International Journal of Intelligent Systems*, 16(9):1053–1067, Sept. 2001.

[29] T. Yang. Computational verb systems: Verb numbers. *International Journal of Intelligent Systems*, 16(5):655–678, May 2001.

[30] T. Yang. *Impulsive Control Theory*, volume 272 of *Lecture Notes in Control and Information Sciences*. Spinger-Verlag, Berlin, Aug. 2001. ISBN 354042296X.

[31] T. Yang. *Computational Verb Theory: From Engineering, Dynamic Systems to Physical Linguistics*, volume 2 of *YangSky.com Monographs in Information Sciences*. Yang's Scientific Research Institute, Tucson, AZ, Oct. 2002. ISBN:0-9721212-1-8.

[32] T. Yang. Computational verb systems: Verbs and dynamic systems. *International Journal of Computational Cognition*, 1(3):1–50, Sept. 2003.

[33] T. Yang. *Fuzzy Dynamic Systems and Computational Verbs Represented by Fuzzy Mathematics*, volume 3 of *YangSky.com Monographs in Information Sciences*. Yang's Scientific Press, Tucson, AZ, Sept. 2003. ISBN:0-9721212-2-6.

[34] T. Yang. *Physical Linguistics: Measurable Linguistics and Duality Between Universe and Cognition*, volume 5 of *YangSky.com Monographs in Information Sciences*. Yang's Scientific Press, Tucson, AZ, Dec. 2004.

[35] T. Yang. Simulating human cognition using computational verb theory.

Journal of Shanghai University(Natural Sciences), 10(s):133–142, Oct. 2004.

[36] T. Yang. Architectures of computational verb controllers: Towards a new paradigm of intelligent control. *International Journal of Computational Cognition*, 3(2):74–101, June 2005 [available online at http://www.YangSky.com/ijcc/ijcc32.htm, http://www.YangSky.us/ijcc/ijcc32.htm].

[37] T. Yang. Applications of computational verbs to the design of P-controllers. *International Journal of Computational Cognition*, 3(2):52–60, June 2005 [available online at http://www.YangSky.us/ijcc/ijcc32.htm, http://www.YangSky.com/ijcc/ijcc32.htm].

[38] T. Yang. Applications of computational verbs to digital image processing. *International Journal of Computational Cognition*, 3(3):31–40, September 2005 [available online at http://www.YangSky.us/ijcc/ijcc33.htm, http://www.YangSky.com/ijcc/ijcc33.htm].

[39] T. Yang. Bridging the Universe and the Cognition. *International Journal of Computational Cognition*, 3(4):1–15, December 2005 [available online at http://www.YangSky.us/ijcc/ijcc34.htm, http://www.YangSky.com/ijcc/ijcc34.htm].

[40] T. Yang. Applications of computational verbs to effective and realtime image understanding. *International Journal of Computational Cognition*, 4(1):49–67, March 2006 [available online at http://www.YangSky.com/ijcc/ijcc41.htm, http://www.YangSky.us/ijcc/ijcc41.htm].

[41] T. Yang. Applications of computational verbs to feeling retrieval from texts. *International Journal of Computational Cognition*, 4(3):28–45, September 2006 [available online at http://www.YangSky.com/ijcc/ijcc43.htm, http://www.YangSky.us/ijcc/ijcc43.htm].

[42] T. Yang. Rule-wise linear computational verb systems: Dynamics and control. *International Journal of Computational Cognition*, 4(4):18–33, December 2006 [available online at http://www.YangSky.com/ijcc/ijcc44.htm, http://www.YangSky.us/ijcc/ijcc44.htm].

[43] T. Yang. Applications of computational verbs to cognitive models of stock markets. *International Journal of Computational Cognition*, 4(2):1–13, June 2006 [available online at http://www.YangSky.us/ijcc/ijcc42.htm, http://www.YangSky.com/ijcc/ijcc42.htm].

[44] T. Yang. Applications of computational verbs to the study of the effects of Russell's annual index reconstitution on stock markets. *International Journal of Computational Cognition*, 4(3):1–8, September 2006 [available online at http://www.YangSky.us/ijcc/ijcc43.htm, http://www.YangSky.com/ijcc/ijcc43.htm].

[45] T. Yang. Bridging computational verbs and fuzzy membership functions using computational verb collapses. *International Journal of Computational Cognition*, 4(4):47–61, December 2006 [available online at http://www.YangSky.us/ijcc/ijcc44.htm, http://www.YangSky.com/ijcc/ijcc44.htm].

[46] T. Yang. Computational verb decision trees. *International Journal of Computational Cognition*, 4(4):34–46, December 2006 [available online at http://www.YangSky.us/ijcc/ijcc44.htm, http://www.YangSky.com/ijcc/ijcc44.htm].

[47] T. Yang. Distances and similarities of saturated computational verbs. *International Journal of Computational Cognition*, 4(4):62–77, December 2006 [available online at http://www.YangSky.us/ijcc/ijcc44.htm, http://www.YangSky.com/ijcc/ijcc44.htm].

[48] T. Yang. Stable computational verb controllers. *International Journal of Computational Cognition*, 4(4):9–17, December 2006 [available online at http://www.YangSky.us/ijcc/ijcc44.htm, http://www.YangSky.com/ijcc/ijcc44.htm].

[49] T. Yang. Using computational verbs to cluster trajectories and curves. *International Journal of Computational Cognition*, 4(4):78–87, December 2006 [available online at http://www.YangSky.us/ijcc/ijcc44.htm, http://www.YangSky.com/ijcc/ijcc44.htm].

[50] T. Yang. Accurate video flame-detecting system based on computational verb theory. AS *Installer*, (42):154–157, August 2007. (in Chinese).

[51] T. Yang. *The Mathematical Principles of Natural Languages: The First Course in Physical Linguistics*, volume 6 of *YangSky.com Monographs in Information Sciences*. Yang's Scientific Press, Tucson, AZ, Dec. 2007. ISBN:0-9721212-4-2.

[52] T. Yang. Applications of computational verb theory to the design of accurate video flame-detecting systems. *International Journal of Computational Cognition*, 5(3):25–42, September 2007 [available online at http://www.YangSky.us/ijcc/ijcc53.htm, http://www.YangSky.com/ijcc/ijcc53.htm].

[53] T. Yang. Cognitive engineering and cognitive industry. *International Journal of Computational Cognition*, 5(3):1–24, September 2007 [available online at http://www.YangSky.us/ijcc/ijcc53.htm, http://www.YangSky.com/ijcc/ijcc53.htm].

[54] T. Yang. Computational verb neural networks. *International Journal of Computational Cognition*, 5(3):57–62, September 2007 [available online at http://www.YangSky.us/ijcc/ijcc53.htm, http://www.YangSky.com/ijcc/ijcc53.htm].

[55] T. Yang. Computational verb theory: Ten years later. *International Journal of Computational Cognition*, 5(3):63–86, September 2007 [available online at http://www.YangSky.us/ijcc/ijcc53.htm, http://www.YangSky.com/ijcc/ijcc53.htm].

[56] T. Yang. Learning computational verb rules. *International Journal of Computational Cognition*, 5(3):43–56, September 2007 [available online at http://www.YangSky.us/ijcc/ijcc53.htm, http://www.YangSky.com/ijcc/ijcc53.htm].

[57] T. Yang. Computational verb rule bases. *International Journal of Computational Cognition*, 6(3):23–34, September 2008 [available online at http://www.YangSky.us/ijcc/ijcc63.htm, http://www.YangSky.com/ijcc/ijcc63.htm].

[58] T. Yang. Simplest computational verb rules and their reasoning. *International Journal of Computational Cognition*, 6(3):35–41, September 2008 [available online at http://www.YangSky.us/ijcc/ijcc63.htm, http://www.YangSky.com/ijcc/ijcc63.htm].

[59] T. Yang. Interactions between computational verbs. *International Journal of Computational Cognition*, 6(4):1–11, December 2008 [available online at http://www.YangSky.us/ijcc/ijcc64.htm, http://www.YangSky.com/ijcc/ijcc64.htm].

[60] T. Yang. Trend-based computational verb similarity. *International Journal of Computational Cognition*, 6(4):24–33, December 2008 [available online at http://www.YangSky.us/ijcc/ijcc64.htm, http://www.YangSky.com/ijcc/ijcc64.htm].

[61] T. Yang and Y. Guo. Measures of ambiguity of computational verbs based on computational verb collapses. *International Journal of Computational Cognition*, 5(4):1–12, December 2007 [available online at http://www.YangSky.us/ijcc/ijcc54.htm, http://www.YangSky.com/ijcc/ijcc54.htm].

[62] Jian Zhang and Minrui Fei. Determination of verb similarity in computational verb theory. *International Journal of Computational Cognition*, 3(3):74–77, September 2005 [available online at http://www.YangSky.us/ijcc/ijcc33.htm, http://www.YangSky.com/ijcc/ijcc33.htm].

[63] Sheng Zhu, Zhong-Jie Wang, Yong Liu, and Bao-Liang Xia. An improvement of the design of computational verb PID-controllers. *System Simulation Technology*, 2(1):25–30, Jan. 2006. (in Chinese).