Time-segmentation- and Position-free Recognition from Video of Air-drawn Gestures and Characters

Yuki Nitsuma, Syunpei Torii, Yuichi Yaguchi and Ryuichi Oka
Image Processing Laboratory, University of Aizu, Ikki-machi Turuga, Fukushima, Japan

Keywords: Gesture Recognition, Segmentation-free Recognition, Position-free Recognition, Moving Camera, Dynamic Programming.

Abstract: We report on the recognition from video streams of isolated alphabetic characters and connected cursive textual characters, such as alphabetic, hiragana a kanji characters, drawn in the air. This topic involves a number of difficult problems in computer vision, such as the segmentation and recognition of complex motion from video. We utilize an algorithm called time-space continuous dynamic programming (TSCDP) that can realize both time- and location-free (spotting) recognition. Spotting means that prior segmentation of input video is not required. Each of the reference (model) characters used is represented by a single stroke composed of pixels. We conducted two experiments involving the recognition of 26 isolated alphabetic characters and 23 Japanese hiragana and kanji air-drawn characters. Moreover we conducted gesture recognition experiments based on TSCDP and showed that TSCDP was free from many restrictions imposed upon conventional methods.

1 INTRODUCTION

Recognition from a video stream of air-drawn gestures and characters will be an important technology for realizing verbal and nonverbal communication in human-computer interaction. However, it is still a challenging research topic involving a number of difficult problems in computer vision, such as segmentation in both time and spatial position, and the recognition of complex motion from a video. According to a survey (Ong and Ranganath, 2005) on gesture and sign language recognition, the following restrictions are necessary for realizing a gesture or sign language recognition system:

- long-sleeved clothing
- colored gloves
- uniform background
- complex but stationary background
- head or face stationary or with less movement than hands
- constant movement of hands
- fixed body location and pose-specific initial hand location
- face and/or left hand excluded from field-of-view
- vocabulary restricted or unnatural signing to avoid overlapping hands or hands occluding face
- field-of-view restricted to the hand, which is kept at fixed orientation and distance to camera

We utilized an algorithm called time-space continuous dynamic programming (TSCDP) (Oka and Matsuzaki, 2012) to be free from these restrictions. TSCDP can realize both position- and segmentation-free (spotting) recognition of a reference point (pixel) trajectory from a time-space pattern such as a video. Spotting means that prior segmentation along the time and spatial axes of the input video is not required. To apply TSCDP, we made a reference model of each character, represented by a single stroke composed of pixels and their location parameters. TSCDP can be applied to two kinds of characters in the air, both isolated and connected characters. Spotting recognition via TSCDP is better suited than conventional methods for recognizing connected air-drawn characters. This is because time segmentation is required to separate connected characters into individual characters, and because position variation can be large when connected characters are drawn in the air. We used a video of air-drawn isolated characters, unadorned with tagging data such as start or end times or location of the characters. To obtain video data for connected characters, we used a famous work from Japanese literature (the "Waka of Hyakunin Isshu"), drawn in the air. We made a set of reference point trajectories, each of which represented a single stroke corresponding to

Nitsuma Y., Torii S., Yaguchi Y. and Oka R..
Time-segmentation- and Position-free Recognition from Video of Air-drawn Gestures and Characters.
DOI: 10.5220/0004816805880599
In *Proceedings of the 3rd International Conference on Pattern Recognition Applications and Methods* (ICPRAM-2014), pages 588-599
ISBN: 978-989-758-018-5

an alphabetic, hiragana or kanji character.

2 RELATED WORK

There has been much research into recognizing air-drawn characters. The projects described below aimed to recognize isolated air-drawn characters, but recognition from a video stream of connected air-drawn characters has not yet been investigated. Okada and Muraoka et al. (Okada and Muraoka, 2003; Kolsch and Turk, 2004; Yang et al., 2002) proposed a method for extracting hand area with brightness values, together with the position of the center of the hand, and evaluated that technique. Horo and Inaba (Horo and Inaba, 2006) proposed a method for constructing a human model from images captured by multiple cameras and obtaining the barycentric position for this model. By assuming that the fingertip voxels would be those furthest from this position, they could extract the trajectory of the fingertips and were then able to recognize characters via continuous dynamic programming (CDP) (Oka, 1998). Sato et al. (Sato et al., 2010) proposed a method that used a time-of-flight camera to obtain distances, extract hand areas, and calculate some characteristic features. They then achieved recognition by comparing reference features and input features via a hidden Markov model. Nakai and Yonezawa et al. (Nakai and Yonezawa, 2009; Gao et al., 2000) proposed a method that used an acceleration sensor (e.g., Wii Remote Controller) to obtain a trajectory which was described in terms of eight stroke directions. They then recognized characters via a character dictionary. Scaroff et al. (Sclaroff et al., 2005; Alon, 2006; Chen et al., 2003; Gao et al., 2000) proposed a matching method for time-space patterns using dynamic programming (DP). Their method used a sequence of feature vectors to construct a model of each character. Each feature vector was composed of four elements, namely the location (x,y) and the motion parameters (v_x,v_y) (more precisely, their mean and variance). Their method therefore requires users to draw characters within a restricted spatial area of a scene. Moreover, movement in the background or video captured by a moving camera is not accommodated, because the motion parameters for the feature vector of the model would be strongly affected by any movement in the input video.

These conventional methods (except for the method of Ezaki et al. (Ezaki et al., 2010), which used an acceleration sensor) use local features comprising depth, color, location parameters, and motion parameters, etc., to construct each character model. They then applied algorithms such as DP or a hidden Markov model to match models to the input patterns. Such methods remain problematic because such local features are not robust when confronted with the demandingly severe characteristics of the real world. For recognizing air-drawn characters, conventional methods perform poorly if there are occlusions, spatial shifting of the characters drawn in the scene, moving backgrounds, or moving images captured by a moving camera.

3 CDP

CDP (Oka, 1998) recognizes a temporal sequence pattern from an unbounded, non-segmented, temporal sequence pattern. TSCDP is a version of CDP that is extended by embedding the space parameter (x,y) into CDP. To show how TSCDP differs from CDP, we first explain CDP. The algorithm in eqn. (3) calculates the optimal value of the evaluation function in eqn. (1). Define a reference sequence $g(\tau), 1 \leq \tau \leq T$ and an input sequence $f(t), t \in (-\infty,\infty)$. Define notations $P = (-\infty,t], Q = [1,T], i = 1,2,\ldots,T, t(i) \in P, \tau(i) \in Q$, a function $r(i)$ mapping from $\tau(i)$ to $t(i)$ and a vector of functions $r = (r(1),r(2),\ldots,r(T))$. There is a constraint between $r(i)$ and $r(i+1)$ as determined by the local constraint of CDP, as shown in Figure 1(a). Then the minimum value of the evaluation function is given by

$$D(t,T) = \min_{r} \sum_{i=1}^{T} \{d(r(i),t(i))\} \qquad (1)$$

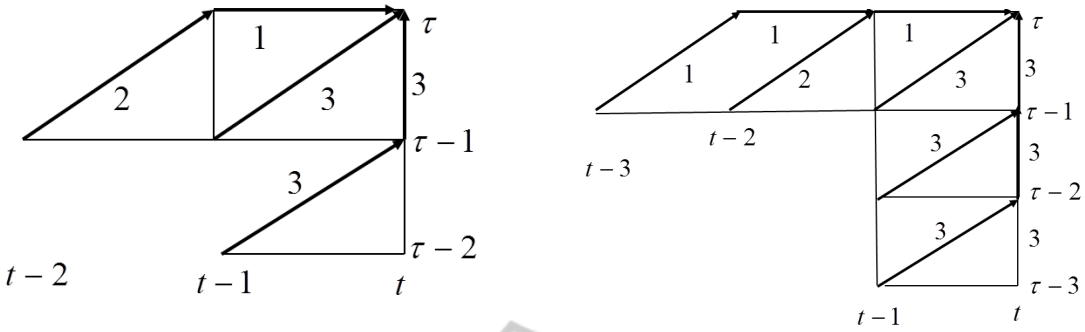
where $t(1) \leq t(2) \leq \cdots \leq t(T) = t$.
The recursive equation in eqn. (3) gives the minimum for the evaluation function in equation (1) by accumulating local distances defined by

$$d(t,\tau) = ||f(t) - g(\tau)||. \qquad (2)$$

The recursive equation for determining $D(t,T)$ is then described by

$$D(t,\tau) = \min \begin{cases} D(t-2,\tau-1) + 2d(t-1,\tau) + d(t,\tau); \\ D(t-1,\tau-1) + 3d(t,\tau); \\ D(t-1,\tau-2) + 3d(t,\tau-1) + 3d(t,\tau). \end{cases} \qquad (3)$$

The boundary condition is $D(t,\tau) = \infty$, $t \leq 0$, $\tau \notin [1,T]$. When accumulating local distances optimally, CDP performs time warping to allow for variation from half to twice the reference pattern. The selection of the best local paths is performed by the recursive equation in eqn. (3). Figure 1 shows two types of local constraints used in CDP for time normalization. In this paper, we use type (a). Other normalization such as from one quarter to four times can be realized in the similar way.



(a) Time normalization from half to twice (b) Time normalization from one third to thrice

Figure 1: Two types of local constraints used in CDP. The number attached to each edge (arrow) indicates weight of the path.

4 TSCDP

We adopted a new concept for devising an algorithm called TSCDP (Oka and Matsuzaki, 2012), which has two advantages. First, the algorithm can perform position-free and robust matching between a reference pattern and an input video captured by a camera in real situations, which might include position shifting, complex or moving backgrounds, and occlusions. Second, the algorithm can also perform both recognition and temporal segmentation in a single process. Therefore, no segmentation is required before applying TSCDP. The original paper of (Oka and Matsuzaki, 2012) proposed this algorithm, but the TSCDP algorithm based on the concept was neither well implemented nor evaluated by applying real motion data. Therefore we conducted both rigorous implementation of the algorithm and experimental evaluation with real data. In addition to isolated air-drawn characters, TSCDP can recognize connected cursive characters, without start and end times having to be specified.

4.1 Evaluation Function for TSCDP

Define a reference time-space pattern, i.e., a pixel-based time-space series over a time interval, by

$$Z(\xi(\tau),\eta(\tau)),\tau = 1,2,\ldots,T, \quad (4)$$

where $(\xi(\tau),\eta(\tau))$ is the location in a two-dimensional plane and Z is the pixel with a gray scale or color value at that location. Define an unbounded input time-space pattern with pixels of gray scale or color values as

$$f(x,y,t),(1 \le x \le M,\ 1 \le y \le N,\ 1 \le t < \infty), \quad (5)$$

where M,N indicate space sizes. Let

$$v_\xi(\tau) = \xi(\tau) - \xi(\tau-1),\ v_\eta(\tau) = \eta(\tau) - \eta(\tau-1) \quad (6)$$

be the respective differences in ξ,η along a reference trajectory. TSCDP does not use parameters ξ,η explicitly in either the solution algorithm or the local distance shown later. In the recursive expression of eqn. (10), difference parameters v_ξ, v_η are used instead of ξ,η. This enables TSCDP to become position-free when recognizing moving objects from a video. The local distance is defined by

$$d(x,y,\tau,t) = \|Z(\xi(\tau),\eta(\tau)) - f(x,y,t)\|. \quad (7)$$

Next, we define the minimum value for the evaluation function. Define the following notations:

$$\begin{aligned} &x(i) \in X,\ y(i) \in Y,\ \xi(i) \in X,\ \eta(i) \in Y, \\ &x = x(T),\ y = y(T), \\ &\tau(T) = T,\ t(T) = t, \\ &\text{a mapping function } \mathrm{u_i} : (\xi(i),\eta(i)) \to (x(i),y(i)), \end{aligned} \quad (8)$$

Let $\mathrm{w} = (r,u_1,u_2,\ldots,u_T)$ be a vector of functions, where a vector of functions r is defined as for CDP. The optimized function is defined by

$$S(x,y,T,t) = \min_w \{\sum_{i=1}^{T} d(x(i),y(i),\tau(i),t(i))\}. \quad (9)$$

4.2 Algorithm for TSCDP

When recognizing isolated or connected air-drawn characters, temporal shrinking and expansion can occur together with spatial shifting. The following formula is the algorithm to determine $S(x,y,T,t)$ by performing time-space warping. The allowable ranges for shrinking and expansion in time and space are each from half to twice the reference point trajectory. Temporal shrinking and expansion from half to twice is realized by the CDP part embedded in TSCDP. Spatial shrinking and expansion from half to double is realized by the second minimum calculation of TSCDP

using a parameter set A. Here, we use a parameter set $\{\frac{1}{2},1,2\}$, which allows spatial shrinking and expansion from half to twice the reference pattern.

$$
\begin{aligned}
&A=\{\frac{1}{2},1,2\},\\
&S(x,y,1,t)=3d(x,y,1,t);\\
&2\leq\tau\leq T:\\
&S(x,y,\tau,t)=\min_{\alpha\in A}\min\\
&\left\{\begin{array}{l}
S(x-\alpha\cdot v_\xi(\tau),y-\alpha\cdot v_\eta(\tau),\tau-1,t-2)\\
+2d(x,y,\tau,t-1)+d(x,y,\tau,t);\\
S(x-\alpha\cdot v_\xi(\tau),y-\alpha\cdot v_\eta(\tau),\tau-1,t-1)\\
+3d(x,y,\tau,t);\\
S(x-\alpha\cdot(v_\xi(\tau)+v_\xi(\tau-1)),\\
y-\alpha\cdot(v_\eta(\tau)+v_\eta(\tau-1)),\tau-2,t-1)\\
+3d(x-\alpha\cdot v_\xi(\tau),\\
y-\alpha\cdot v_\eta(\tau),\tau-1,t)+3d(x,y,\tau,t).
\end{array}\right. \qquad (10)
\end{aligned}
$$

The boundary condition is defined by

$$
\begin{aligned}
&S(x,y,\tau,t)=\infty,\ d(x,y,\tau,t)=\infty,\\
&\text{if }\ (x,y)\notin[M,N],\ t\leq 0,\tau\notin[1,T].
\end{aligned}
$$

We explain the basic mechanism of the local computation of TSCDP. Eqn. (10) below works for time-space optimization of the evaluation shown by eqn. (9) as illustrated in Figure 2. The function of the time normalization part of eqn. (10) is the same as that for CDP. The function for space normalization is simply added to CDP by introducing (x,y)-space to (t,τ) space. That is, we consider an algorithm working in a four-dimensional space, (x,y,t,τ). There are three candidate paths in the scheme of CDP. Space normalization is realized by through each of three paths of CDP. The simplest case is the third path shown in Figure 2. The third path of TSCDP includes the third path of CDP in the four-dimensional space. The two times t and $t-1$ appearing in the third path of CDP have three points of τ, namely $\tau,\tau-1,\tau-2$. Therefore we can consider three points in 4D space. Then the locations of (x,y)-coordinates each from the three points have τ parameters, respectively. We consider that each difference between them of the τ parameter corresponds to each difference of (x,y) of images in the input video. The difference of (x,y) is represented by using (v_ξ,v_η), as shown in Figure 2. Then we embed the suitable parameters of (v_ξ,v_η) into $S(x,y,\tau,t)$ and $d(x,y,\tau,t)$ of eqn. (10). In this situation, only space parameter (x,y) is embedded. No space normalization is realized. If the size of (v_ξ,v_η) is modified, space normalization of the reference pattern can be realized. Now we consider three types of space size modification at each local optimization, namely $\{\frac{1}{2},1,2\}$. This means that any combination of local spatial size modifications from half to twice the reference pattern can be realized. This function is realized by introducing parameters, α, $A=\{\frac{1}{2},1,2\}$, and $\min_{\alpha\in A}$ into the recursive TSCDP equation. The first and second candidate paths of eqn. (10) are handled in a similar way to the third.

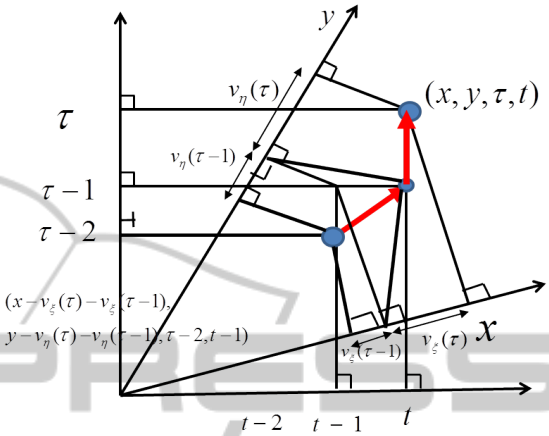


Figure 2: Eqn. (10) realizes an optimal pixel matching using three candidate paths for time normalization by accumulation of local distance between pixel values of a reference and an input video. The figure shows how the third path works during optimal path selection in 4D space. The other two paths work in a similar way.

Eventually, the allowable time-space search arrives at the time-space point (x,y,T,t), where the optimal matching trajectory is determined by TSCDP, as shown in Figure 4 for the reference pattern shown in Figure 3. The allowable search area is dependent on the reference model for the pixel sequence. That is, each reference model has its own allowable search area. This differs from conventional DP matching algorithms, which have the same search areas for all reference sequences.

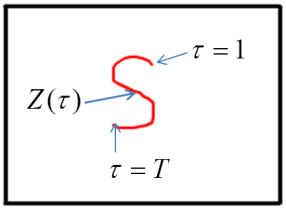


Figure 3: A reference pattern (pixel sequence) made by drawing one stroke sequence (a sequence of location parameters, $((\xi(\tau),\eta(\tau)),\tau=1,2,...,T)$ on a two-dimensional plane, where the length of stroke correspond to T) of the reference pattern. Each pixel value $Z(\xi(\tau),\eta(\tau))$ of the reference pattern is assigned a constant skin color.

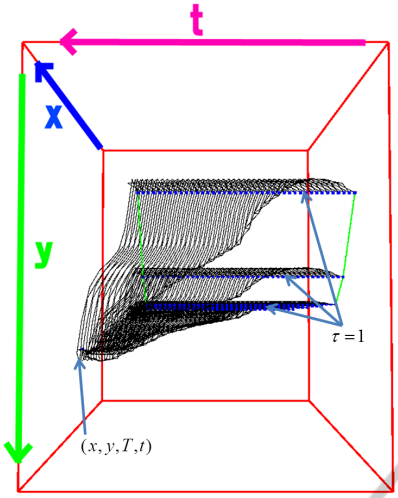


Figure 4: Search space for TSCDP in arriving at the time-space point (x,y,T,t). Each reference model has its own search space.

4.3 Time-segmentation-free and Position-free Recognition

Time-space spotting recognition of a reference pattern from an input video is determined frame by frame by formula (11), using [local area], which refers to an area composed of locations satisfying $S(x,y,T,t) \leq h$ (where h is a given threshold value). The optimal accumulated value $S(x,y,T,t)$ at each time t indicates a two-dimensional scalar field with respect to (x,y). Location (x,y) is called *a recognition location* if it satisfies the condition $S(x,y,T,t) \leq h$. A recognition location indicates that a category represented by a reference pattern is recognized at time $t \in [0,T]$ and location (x,y). Usually, locations neighboring a recognition location are also recognition locations, because these locations have similar matching trajectories in the 4D $((x,y,\tau,t))$ space. We call such locations *the local area of recognition locations*.

At each time t, we can find an arbitrary number of local areas of recognition locations, depending upon the number of existing time-space patterns, which are optimally matched with a reference pattern. Then we can determine a location, denoted by (x^*,y^*), giving the minimum value of $S(x,y,T,t)$ for each local area of recognition locations. The number of these locations is the number of recognition locations at time t. A local area of recognition locations can be created at an arbitrary position on the (x,y)-plane depending on the query video. This procedure based on $S(x,y,T,t)$ is the realization of the position-free (spotting) recognition of TSCDP. On the other hand, a local minimum location (x^*,y^*) has time parameter t. If we consider a time series of a local minimum location, we can detect a time duration, denoted by $[t_s,t_e]$, satisfying $S(x^*,y^*,T,t) \leq h, t \in [t_s,t_e]$. The minimum value, denoted by $S(x^*,y^*,T,t_{\mathrm{reco}})$, among $S(x^*,y^*,T,t)$, $t \in [t_s,t_e]$, corresponds to the recognition considering time-space axes. The time t_{reco} indicates the end time of a recognized pattern in an input query video determined without any segmentation in advance. The starting time of the recognized pattern is determined by back-tracing the matching paths of TSCDP, starting from t_{reco}. This procedure is the realization of time-segmentation-free (spotting) recognition based on TSCDP. The following are the algorithms for the above procedures. The term [local area] in the following formulae is *the local area of recognition locations* in the above discussion.

$$(x^*,y^*,T,t) = arg \min_{(x,y)\in[\text{local area}]} \{S(x,y,T,t)\} \quad (11)$$

Spotting recognition for multiple categories is determined by using multiple reference patterns. Define the i-th reference pattern of a pixel series that corresponds to the i-th category by

$$Z_i(\xi(\tau),\eta(\tau)), \tau = 1,2,\ldots,T_i. \quad (12)$$

TSCDP then detects one or more $S_i(x^*,y^*,T_i,t)$ values as frame-by-frame minimum accumulation values for which $\frac{S_i(x^*,y^*,T_i,t)}{3T_i} \leq h$ is satisfied. The following equations determine the spotting result for multiple categories:

$$i^*(t) = arg \min_i \frac{S_i(x^*,y^*,T,t)}{3T_i} \quad (13)$$

$$S_i(x^*,y^*,T_i,t) = \min_{(x,y)\in[\text{local area}]} S_i(x,y,T,t). \quad (14)$$

Figure 5 shows the time-segmentation-free (spotting) recognition of connected cursive air-drawn characters.

5 RECOGNIZING ISOLATED AND CONNECTED AIR-DRAWN CHARACTERS

The first step in recognizing air-drawn characters via TSCDP is to make a model of each character category as a reference pattern for TSCDP. Such a TSCDP reference pattern (model) is determined by a stream of pixels forming a trajectory on a two-dimensional plane. This procedure corresponds to a learning procedure for making a model used in conventional on- or off-line character recognition. But our method is different from conventional learning methods. We do not use sample videos for making reference patterns for TSCDP. A reference pattern of

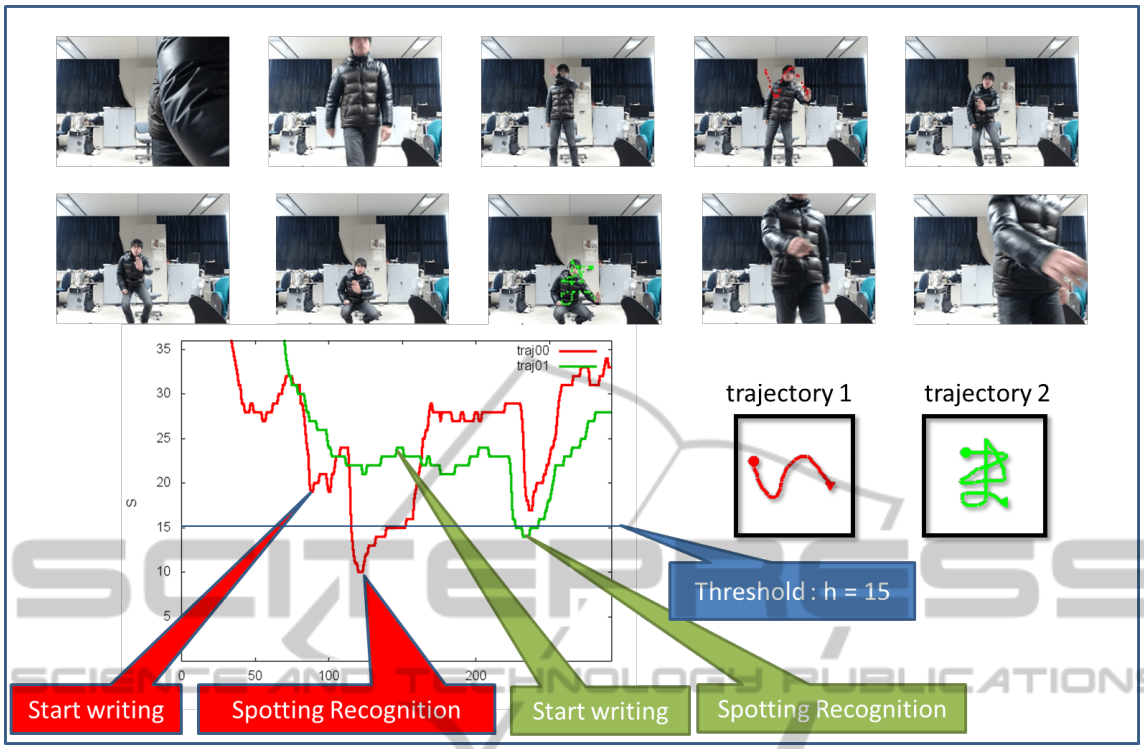


Figure 5: Time-segmentation-free recognition for connected characters. Two connected characters are separately recognized at different time points without any advance segmentation.

TSCDP is made by air-drawing one stroke projected on a two-dimensional plane. A stroke is a sequence of pixel location parameters $\xi(\tau), \eta(\tau)$, $\tau = 1, 2, ..., T$. The length of stroke corresponds to T. Each location of the stroke is assigned a pixel value, denoted by $Z(\xi(\tau), \eta(\tau))$, expressing a constant skin color. Finally, a reference pattern is represented by $Z(\xi(\tau), \eta(\tau)), \tau = 1, 2, ..., T$. The second step is treatment of single-stroke representation of a model. The stream is composed of connected straight or curved lines. Categories for characters such as 'C,' 'O,' and 'Q' are easy to represent representing a one-stroke model. However, most other characters, including those from the alphabet or Japanese hiragana or kanji characters, cannot be drawn as a single stroke. For these characters, we make a one-stroke model for each character by connecting its separate strokes with additional strokes in the air. These additional strokes are not part of the actual strokes belonging to the character itself. Using this kind of modeling for each character allows TSCDP to be adopted for their recognition.

We prepared single-stroke models for each category of alphabetic and Japanese hiragana and kanji characters. The input pattern is obtained from a video capturing isolated characters or a sequence of connected cursive characters drawn by human hand in the air. We do not specify start and end times for each drawing, even for isolated characters. Furthermore, neither a color finger cap, nor gloves, nor any special device is required. Applying TSCDP to a character model with category number i, we obtain $i^*(t)$, where the time t is called the spotting time. If multiple $i^*(t)$ values are detected, the output for the time is determined by selecting the category with the maximum stroke length.

6 EXPERIMENTS

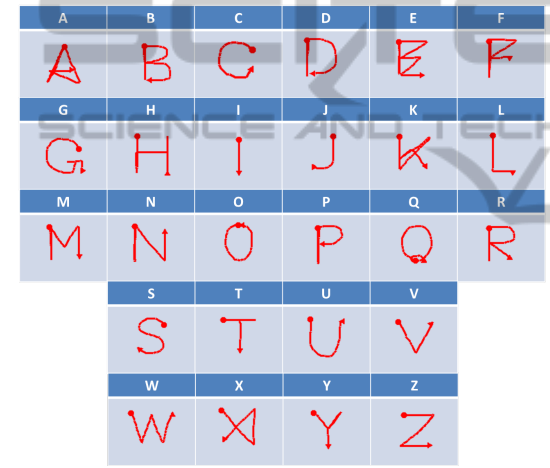
6.1 Database and Performance for a Comparison Study

We use videos obtained by capturing air-drawn gestures and characters made with one stroke in a position-free style. Some of these gesture videos include large occlusion, multiple gestures in a single scene, or connected characters. Some were captured by a moving camera with moving backgrounds. There had been no experiment in the past applying conventional methods to such real data. Therefore, it seems impossible to compare our method with the conventional methods described in such papers (Okada and

Muraoka, 2003; Horo and Inaba, 2006; Sato et al., 2010; Nakai and Yonezawa, 2009). Moreover our database is rather small. Therefore the experiments in this paper are regarded as preliminary trials to investigate whether or not TSCDP can loosen the many constraints mentioned in the introduction, before applying a large amount of real world data. For recognizing air-drawn characters, we apply two kinds of spotting recognitions using the same TSCDP. Final decision algorithms are different from each other, as mentioned in section 4.

6.2 Experimental Conditions

Figure 6(a) shows a set of reference patterns for an alphabet of 26 categories, each of which is a one-stroke model. In addition, we manually constructed a set of



(a) Alphabet

(b) Hiragana & kanji

Figure 6: List of reference patterns. Each reference pattern is made with a single stroke.

one-stroke characters, as seen in Figure 6(b). These one-stroke characters are used in the Waka poem and are to be regarded as the reference patterns when applying TSCDP in parallel. Figure 7 shows a sheet of paper upon which the famous Japanese "Waka Imakomuto" from the "Hyakunin Isshu" is written. We showed this example to the participants, who were instructed to write the Waka in the air using connected characters. The experimental conditions were as follows.

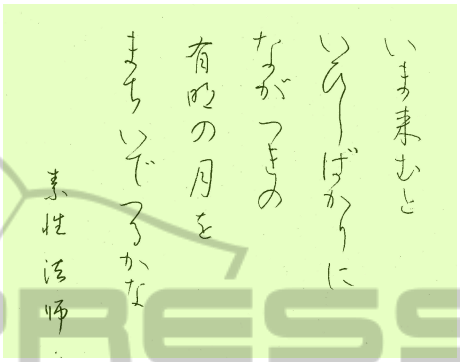


Figure 7: Waka shown to participants drawing a sequence of connected cursive characters in the air.

- Video:
  - Frames per second: 20 fps
  - Resolution: 200 × 150 pixels
  - RGB color was used (8 bits per color)
- Reference patterns for characters:
  - A single stroke reference pattern was constructed manually for each character.
  - A spatial distance of 3 pixels along a stroke in the plane (ξ, η) corresponding to 50 ms in parameter τ of Z. These parameters were fixed for all reference patterns. The total length L pixels of each stroke determined $T = (L/3) \times 50$ ms in Z.
- Scene:
  - Each participant wrote isolated characters in the air without start or end time being specified. They also wrote connected characters in the air, column by column, without a specified start or end time.
- There were three participants.
- The list of hiragana and kanji models (26 alphabetic characters) for recognizing isolated characters is shown in Figure 6(a).
- The list of models (23 references) for recognizing connected characters is shown in Figure 6(b). A writing style for connected cursive characters (Figure 7) was shown to each person in advance.

- Parameters:
  - The spotting recognition threshold was $h = 15$ (fixed).
  - Z = (R, G, B) = (190, 145, 145) (fixed).
  - The Euclidian norm was used for calculating local distance.

6.3 Constraint-free Characteristics of TSCDP

As mentioned in the introduction, most conventional methods for realizing recognition systems are subject to many restrictions. A system based on TSCDP can dispense with many of them, as our experimental results indicate. We did not use long-sleeved clothing or colored gloves. Using a reference pattern composed of pixels with a constant skin color, TSCDP optimally matches only an existing pixel sequence in an input video without identifying any areas of hand or finger. The inferred skin tone is roughly determined, without deep investigation, but TSCDP works well using the heuristically derived skin color. TSCDP seems robust to variation of skin color, and is also robust when faced with complex and moving backgrounds because TSCDP matches only with a sequence of pixels (a macroscopic and specified motion with time length T), so that moving backgrounds, including head or face movement, do not interfere with the total accumulation value of local distances if moving T backgrounds do not have a similar motion to a reference pattern with a period of around length T. Figure 8 illustrates the recognition of a gesture in a complex and moving background.

TSCDP allows non-linear variations from half to twice the velocity of movement by the CDP part of TSCDP. Figure 5 illustrates time-segmentation recognition without any segmentation of start and end time after adapting non-linear time variations. Constraints of fixed body location and pose-specific initial hand location are required by conventional methods because they are position-dependent when they match a model sequence and a video. The model of conventional methods is made by features including location parameters, so that whole target matching procedures are still location-dependent. A reference pattern of TSCDP also has location parameters. However the dependency of location is relaxed by directly embedding the location difference $u_\xi(\tau), v_\eta(\tau)$ into time-warping candidate paths. The position-free characteristic property is then realized in TSCDP, as mentioned in §4.3. The allowance for spatial shrinking and expansion is also realized by embedding path selection for contracting and dilating spatial size using both $u_\xi(\tau), v_\eta(\tau)$ and a set of parameters A in the recursive equation of TSCDP. Figure 9 shows a reference pattern that is recognized at different positions when multiple and similar time-space patterns exist in a video.

TSCDP also is robust when presented with overlapping hands or occlusion, because these cases increase only a relatively small part of accumulated value $S(x,y,T,t)$, depending on the spatial and temporal size of overlapping hands or hands over face or occlusion by objects between the camera and subject. Figure 10 shows how a gesture is correctly recognized even in the presence of occlusion. A reference pattern

Figure 10: The upper the figure shows the case in which the occlusion occurs at the beginning period in drawing the 'S' character. The lower the figure shows the case in which the occlusion occurs during the middle period. TSCDP recognizes character gestures correctly for both cases.

can be made by any kind of single-stroke sequence projected on a two-dimensional plane. Therefore, a reference pattern with complex shape and long duration is acceptable. Chinese kanji characters belong to this category. It becomes even easier to recognize complex and long reference patterns by TSCDP because they are more distinguishable from one another. Complex reference patterns allow a large vocabulary. Figure 11 shows recognition of complex Chinese characters including a character "kuru" ("come" in English), which is the last one in Figure 6(b). A set of gesture patterns caused by various fields-of-view of a hand is generated by non-linear time and space deformations of the reference pattern.

Let $\{F(x,y)|(x,y) \in R\}$ be an image of object at a fixed time t_0, where R is a raster (two-dimension pixel area) and define $t \geq t_0$ as a time. Define a dis-

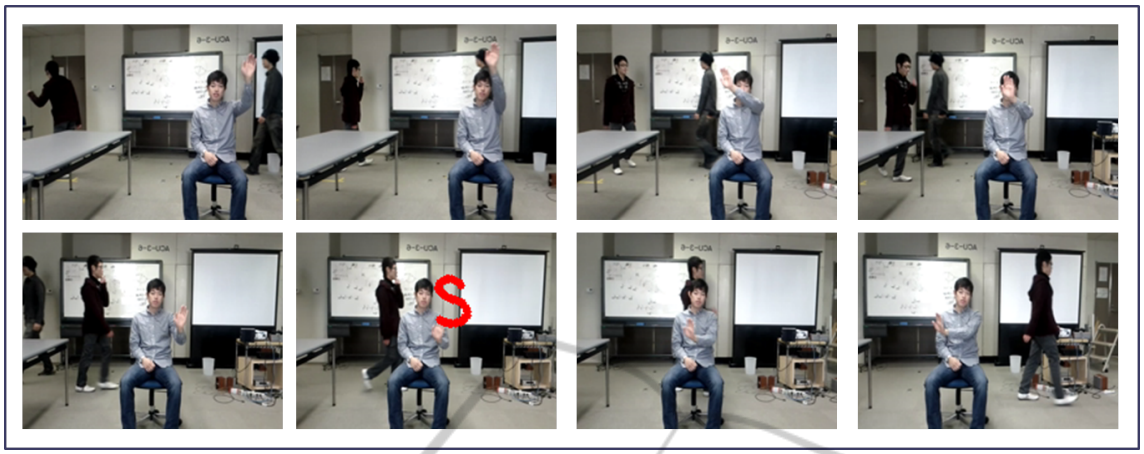


Figure 8: Recognition of an air-drawn gesture under situation the condition of a complex scene with moving objects in the background. The scene includes people walking in the background, and the static background is a normal office environment.

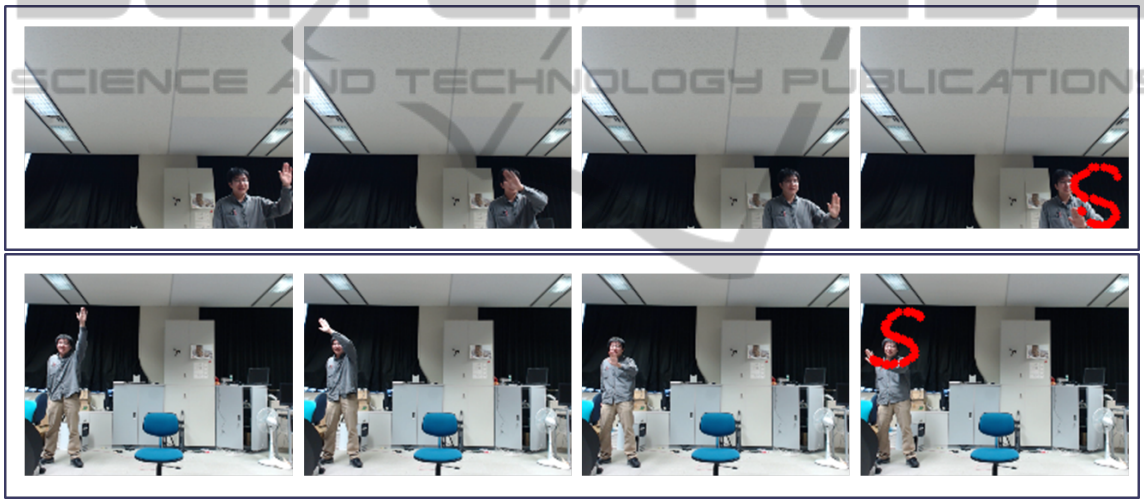


Figure 9: A reference pattern is recognized at different positions (right and down, left and up positions), each of which corresponds to an existing similar trajectory in a video.

Figure 11: Complex Chinese characters are recognized by TSCDP.

tance $p(t)$ [cm] between a camera and an object, and a parameter c whose value is determined by calibration. If the camera moves $p(t)$ forward or backward relative to the object, then the two-dimensional image of an object shrinks or expands, which simple geometric property is described by $\{F(x \times c\frac{p(t_0)}{p(t)}, y \times c\frac{p(t_0)}{p(t)}) | (x,y) \in R\}$, where c is regarded as a parameter to transform a value of distance ratio to pixel size. If the conditions of range $\frac{x}{2} \leq x \times c\frac{p(t_0)}{p(t)} \leq 2x$ and $\frac{y}{2} \leq y \times c\frac{p(t_0)}{p(t)} \leq 2y$ are satisfied, the space normalization of TSCDP works well.

On the other hand, let define $x(t)$ a pixel size of rightward or leftward motion of the camera at time t from $x(t_0) = 0$, where $x(t) > 0$ for rightward motion and $x(t) < 0$ for leftward motion, assuming no vertical movement. Then the two-dimensional image of an object expands to right or left direction and is described by $\{F(x+x(t), y) | (x,y) \in R\}$. If the condition of range $\frac{x}{2} \leq x(t) \leq 2x$ is satisfied, then the space normalization of TSCDP works well.

Let $F(t)$ be an image of the object at t with simultaneous combination of two kinds of a camera motion. Then $F(t)$ is determined by:

$$F(t) = \{F((x+x(t)\}) \times c\frac{p(t_0)}{p(t)}, y \times c\frac{p(t_0)}{p(t)}) | (x,y) \in R\}.$$

If the conditions of the range, $\frac{x}{2} \leq (x + x(t)\}) \times c\frac{p_0}{p(t)} \leq 2x$ and $\frac{y}{2} \leq y \times c\frac{p(t_0)}{p(t)} \leq 2y$, are satisfied, the space normalization of TSCDP well works.

If time shrinking or expansion occurs as a side effect of camera motion, time normalization of TSCDP works well, scaling from half to twice. This reasoning is equivalent to saying that if a pixel trajectory is included in $F(t), (t \in [t_0, t])$ and also belongs to the time-space area of Figure 5 of a reference pattern, then the pixel trajectory is well recognized by TSCDP. Otherwise the accumulated local distance S increases depending on the size of the part of trajectory out of the time-space area of Figure 5. If the increased accumulated distance S is smaller than threshold h, then the trajectory is well recognized. If we set a higher threshold value h, the recognition system becomes more robust to largely deformed input patterns at the cost of increasing error rate. Robustness and error rate are traded-off in determining the threshold value h.

Figure 12 shows that a continuously deforming image of a gesture is well recognized in a video which captures the gesture under changing distance from and orientation to the camera.

6.4 Experimental Results for Isolated Character Recognition

For recognition of isolated characters by TSCDP, there were two types of errors, confused recognition and missing recognition. Missing recognition can occur because, even for isolated characters, a threshold value is used to determine whether or not a spotting output is obtained. If all TSCDP output values $S_i(x^*, y^*, T_i, t)/(3T_i)$ are above the threshold value h for $i = 1, 2, ..., N$, and $t \in (-\infty, +\infty)$, there will be no output recognition from the input video. The recognition rates are shown in Table 1.

Table 1: Results of the first experiment.

Result	Total	Subject A	Subject B	Subject C
Correct	62.6%	46.2%	61.5%	80.1%
Missing	5.1%	0.0%	3.8%	11.5%
Confusion	32.3%	33.8%	34.7%	18.4%

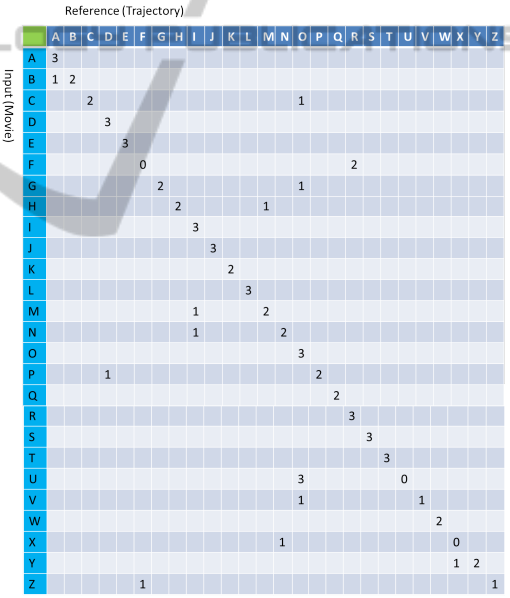


Figure 13: Confusion matrix of recognition results for isolated characters.

Figure 13 shows the confusion matrix for the recognition of isolated characters.

6.5 Experimental Results for Connected Character Recognition

Researching recognition of connected characters by TSCDP, there were three types of errors. The first, "missing (M)," means that no category was detected at the correct time. The second, "ghost (G)," means that an output appeared at an incorrect time. The third,

(a) starting image of the moving camera. (b) ending image of the moving camera.

Figure 12: A moving camera captures a deforming gesture caused by different distance and orientation to the camera. TSCDP can recognize the deforming gesture.

"confusion (F)," means that a category was detected at the correct time but the category was incorrect. Correct output, "correct (C)," means that correct output was obtained at the correct time. We can then determine each recognition rate as follows.

- Correct rate = $\frac{C}{(M+G+F+C)} \times 100\%$
- Missing rate = $\frac{M}{(M+G+F+C)} \times 100\%$
- Ghost rate = $\frac{G}{(M+G+F+C)} \times 100\%$
- Confusion rate = $\frac{F}{(M+G+F+C)} \times 100\%$

The recognition rates are shown in Table 2, where Ghost rate = 0%, and the confusion matrix is shown in Figure 14.

Table 2: Results of the second experiment.

Result	Total	Subject A	Subject B	Subject C
Correct	64.4%	82.8%	62.1%	48.3%
Missing	11.1%	3.4%	17.2%	13.8%
Confusion	24.5%	13.8%	20.7%	37.9%

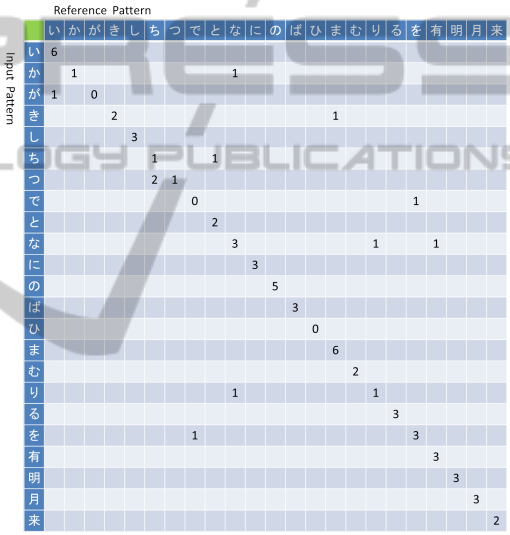


Figure 14: Confusion matrix of recognition results for connected alphabetic hiragana and kanji characters.

7 CONCLUSIONS

This study confirmed that TSCDP can work well for recognizing both isolated and connected cursive air-drawn characters from a video. In particular, connected air-drawn characters can be recognized without time-segmentation in advance. Moreover, we presented several experimental results of gesture recognition that demonstrate how TSCDP is free from many constraints, including position restrictions, that are imposed by conventional methods for realizing a recognition system of gesture or sign language.

REFERENCES

Alon, J. (2006). *Spatiotemporal Gesture Segmentation*. Dissertation for Doctor of Philosophy, Boston University.

Chen, F., Fu, C., and Huang., C. (2003). Hand gesture recognition using a real-time tracking method and hidden markov models. *Image and Video Computing*, 21(8):745–758.

Ezaki, N., Sugimoto, M., Kiyota, K., and Yamamoto, S. (2010). Character recognition by using acceleration sensor: Proposing a character input method using wiimote [in Japanese]. *Meeting on Image Recognition and Understanding*, IS2–48:1094–1098.

Gao, W., Ma, J., J.Wu, and Wang, C. (2000). Sign language recognition based on hmm/ann/dp. *International Journal of Pattern Recognition and Artificial Intelligence*, 14(5):587–602.

Horo, T. and Inaba, M. (2006). A handwriting recognition system using multiple cameras [in Japanese]. *Workshop on Interactive Systems and Software (WISS2006)*.

Kolsch, M. and Turk, M. (2004). Robust hand detection. *In Proc. Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 614–619.

Nakai, M. and Yonezawa, H. (2009). Aerial handwritten character recognition using an acceleration sensor [in Japanese]. *Forum on Information Technology*, H–19:133–134.

Oka, R. (1998). Spotting method for classification of real world data. *The Computer Journal*, 41(8):559–565.

Oka, R. and Matsuzaki, T. (2012). Robustness for time-spatial deformation of an occlusion realized in time-space continuous dynamic programming [in Japanese]. *Joint Technical Meeting on Information Processing and Innovative Industrial Systems*, 27(6):873–891.

Okada, T. and Muraoka, Y. (2003). Letter input system for handwriting gestures [in Japanese]. *Transactions of the Institute of Electronics, Information and Communication Engineers*, D-II J86-D-II(7):1015–1025.

Ong, S. C. W. and Ranganath, S. (2005). Automatic sign language analysis: A survey and the future beyond lexical meaning. *Pattern Analysis and Machine Intelligence*, 27(6):873–891.

Sato, A., Shinoda, K., and Furui, S. (2010). Sign language recognition using time-of-flight camera [in Japanese]. *Meeting on Image Recognition and Understanding*, IS3–44:1861–1868.

Sclaroff, S., Betke, M., Kollios, G., Alon, J., Athitsos, V., Li, R., Magee, J., and Tian, T.-P. (2005). Tracking analysis and recognition of human gestures in video. *ICDAR: Int. Conf. on Document Analysis and Recognition*.

Yang, M., Ahuja, N., and Tabb, M. (2002). Extraction of 2d motion trajectories and its application to hand gesture recognition. *Pattern Analysis and Machine Intelligence*, 24(8):1061–1074.