Learning Non-Markovian Constraints for Handwriting Recognition

Ryosuke Kakisako, Seiichi Uchida, Frinken Volkmar
Kyushu University, Fukuoka, Japan 819–0395
Email: uchida@ait.kyushu-u.ac.jp

***Abstract*—Recently, the horizon of dynamic time warping (DTW) for matching two sequential patterns has been extended to deal with non-Markovian constraints. The non-Markovian constraints regulate the matching in a wider scale, whereas Markovian constraints regulate the matching only locally. The global optimization of the non-Markovian DTW is proved to be solvable in polynomial time by a graph cut algorithm. The main contribution of this paper is to reveal what is the best constraint for handwriting recognition by using the non-Markovian DTW. The result showed that the best constraint is not a Markovian but a totally non-Markovian constraint that regulates the matching between very distant points; that is, it was proved that the conventional Markovian DTW has a clear limitation and the non-Markovian DTW should be more focused in future research.**

I. Introduction

Dynamic time warping (DTW) is a classical but still popular technique for temporal pattern recognition, such as recognition of handwriting, gesture, and speech. In general, DTW is formulated as an optimization problem of a nonlinear matching between two temporal patterns and then solved by dynamic programming (DP) [1]. Solution by DP has many merits, such as global optimality of the solution, numerical stability, versatility of its objective function, and computational efficiency.

On the other hand, DP-based solution limits the flexibility of DTW. Specifically, since DP is based on Markovian optimization process, the DTW problem also should assume a Markovian property. This means that we can regulate the matching only locally. In other words, it is impossible to regulate the matching in a wider scale.

Recently, a graph cut-based DTW solution has been proposed [2] and then extended to deal with non-Markovian constraints [3], [4]. Fortunately, graph cut provides the polynomial-time solution for the globally optimal solution even with the non-Markovian constraints.

The main contribution of this paper is to understand what is the best constraint for DTW on a specific pattern recognition problem, i.e., handwriting recognition. This is important because most researches have been done by DTW (and its stochastic extension, HMM) with Markovian constraints, without any skeptic consideration. In other words, although the conventional Markovian constraint often works sufficiently, nobody knows it is the best or not. It is, therefore, very worthy to reveal the best constraint by using the above non-Markovian DTW, which includes the conventional Markovian DTW as its special case.

For this purpose, we try to find the optimal constraint via an automatic learning process. However, learning non-Markovian constraints (or even Markovian constraints) is still an open problem. In fact, the Markovian constraints of Markov random fields (MRF) are often pre-determined manually as 4-neighborhood or 8-neighborhood. Conditional random fields (CRFs) employ non-Markovian constraints but the constraints are also manually designed [5] or even fully-connected [6], [7]. In [8], a simple learning method is employed, where all possible non-Markovian constraints are examined in a one-by-one manner; if the addition of a non-Markovian constraint provides a positive effect on the recognition accuracy, the constraint is accepted.

Since our purpose is to reveal the best constraint for DTW, we will take a brute-force learning strategy at the cost of computational complexity. Then, we will inspect the learned constraints and the reason why those constraints are learned as the best. We also conduct a recognition experiment to measure the usefulness of the best constraint. This experiment will tell us that a non-Markovian constraint for regulating the relationship between long distant points is the best one although it is very different from the conventional Markovian constraint.

II. Fundamental of dynamic time warping

A. Problem Formulation

For two sequential patterns $X = x_1, \ldots, x_t, \ldots, x_T$ and $Y = y_1, \ldots, y_\tau, \ldots, y_\mathcal{T}$, a mapping between them is represented as $U = u_1, \ldots, u_t, \ldots, u_T$, where $u_t \in [1, \mathcal{T}]$ specifies the correspondence between t and τ. For example, if $u_4 = 3$, x_4 corresponds y_3. The mapping U can be illustrated as a path on the t–τ plane as shown in Fig. 1 (a). The path is often called *warping function* or *warping path*.

In general, DTW is formulated as the optimization of U for minimizing the difference between X and Y under the correspondence by U. More formally, DTW is the minimization problem of the following objective function F with respect to $U = u_1, \ldots, u_T$:

$$\min F = \min_{u_1,\ldots,u_T} \sum_{t=1}^{T} d_t(u_t), \tag{1}$$

where $d_t(\tau)$ is so-called *local cost* for evaluating the difference between x_t and y_τ. A typical choice of the local cost is $d_t(\tau) = \|x_t - y_\tau\|$.

During the optimization of F, we often consider the constraint on U. In the conventional DTW, the constraint is defined in the following form:

$$\alpha_1 \leq u_{t+1} - u_t \leq \beta_1, \qquad \forall t, \tag{2}$$

where α_1 and β_1 are constants specifying the minimum and the maximum slope of the warping path. Fig. 2 (a) shows the case where $\alpha_1 = 1$ and $\beta_1 = 3$. In this case, τ is increased by 1 or

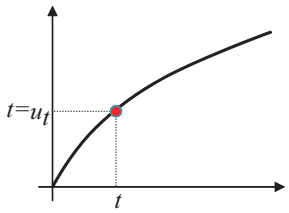

Fig. 1. The mapping U as a warping path on the t–τ plane.

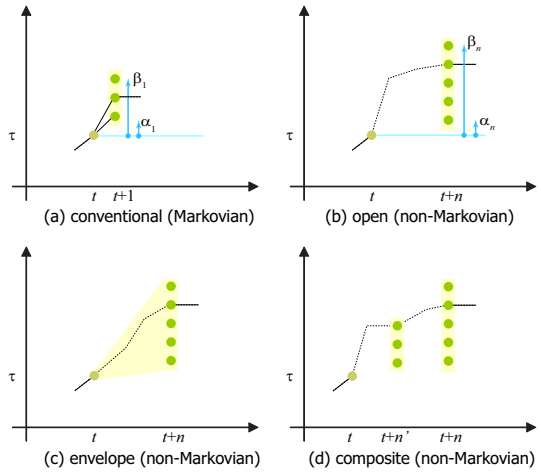

Fig. 2. Constraint types.

2 or 3, when t is increased by 1. In the most typical case, we set $\alpha_1 = 0$ and $\beta_1 = 2$, which is the so-called monotonicity and continuity constraint. This is because the warping path increases monotonically by non-negative α_1 and continuously by non-large β_1. In addition to the above slope constraint (2), boundary conditions

$$u_1 = 1, \qquad u_T = \mathcal{T} \tag{3}$$

are employed for corresponding x_1 to y_1 and x_T to $y_\mathcal{T}$.

The following two points about the constraint (2) are worthy to be noted. First, the constraint (2) is a *Markovian constraint*, as shown in Fig. 2 (a). That is, it regulates the relationship between two neighboring points, t and $t+1$. Second, the constants α_1 and β_1 are integer-valued because t and τ are also integer-valued. This suggests that the Markovian constraint in (2) only can control the minimum and maximum slope in a discrete way. In other words, if we consider the slope of the warping path by $d\tau/dt = u_{t+1} - u_t$, the slope range cannot be specified by an arbitrary value, such as $0.9 \le d\tau/dt \le 1.1$.

B. Solution by Dynamic Programming

The globally optimal solution of the minimization problem (1) subject to the Markovian constraint (2) and the boundary conditions (3) can be obtained by DP. In the solution, we calculate the *DP recursion*

$$f_t(u_t) = d_t(u_t) + \min_{\alpha_1 \le u_t - u_{t-1} \le \beta_1} f_{t-1}(u_{t-1}). \tag{4}$$

from $t = 2$ to T for all $u_t \in [1, \mathcal{T}]$ with the initial value at $t = 1$,

$$f_1(u_1) = \begin{cases} d_1(u_1) & \text{for} u_1 = 1 \\ \infty & \text{otherwise.} \end{cases}$$

Then $\min F = f_T(\mathcal{T})$ and the globally optimal U is obtained by *backtracking* operation that checks u_{t-1} giving the minimum in (4), from $t = T$ to 1.

C. Solution by Graph Cut

The optimization problem of II-B also can be solved by graph cut [2], [3]. Fig. 3 outlines the solution. The solution starts from the creation of a directed graph (b) whose cut corresponds to a warping path of the t–τ plane (a). Most edges except for the black edges have an infinite cost and thus cannot be cut. The black edge corresponding to the grid (t, τ) will have a local cost $d_t(\tau)$. Thus, the minimum-cost cut of (b) has the same minimum cost $\min F$ of the original DTW problem. For further details of this graph cut solution, refer to [3].

III. Non-Markovian Constraints

A. Definition of Non-Markovian Constraints

The non-Markovian constraint is defined as follows:

$$\alpha_n \le u_{t+n} - u_t \le \beta_n. \tag{5}$$

Its difference from the Markovian constraint (2) is to specify a relationship between two distant points, t and $t+n$, directly. The constants α_n and β_n have a subscript n for clarifying that they are the constants for the constraints between t and $t+n$. It is possible to call (5) as a "higher-order" Markovian constraint. We, however, call it the non-Markovian constraint. This is because our constraint specifies a relationship between two distant points, t and $t+n$, and is totally independent of their intermediate points, $t+1, \ldots, t+n-1$. It will be noteworthy that the non-Markovian constraint can realize a fine control of the path slope. For example, the constraint $9 \le u_{t+10} - u_t \le 11$ realizes $0.9 \le d\tau/dt \le 1.1$ approximately.

B. Solution by Graph Cut

Under the non-Markovian constraint (5), the minimization of (1) can no longer be solved by DP in practice. Strictly speaking, it is still possible to solve the problem by DP if we introduce an n-dimensional vector $(u_t, \ldots, u_{t+n-1})$ as a new variable to be optimized at t. However, this remedy is far less practical especially for a larger n because the computational complexity increases from $O(T \cdot \mathcal{T})$ to $O(T \cdot \mathcal{T}^n)$.

Fortunately, the solution by graph cut in II-C still can provide the globally optimal solution under the non-Markovian constraint with polynomial-time computations. Just by adding the edges representing the long-distant constraint of Fig. 3(c), we obtain the warping path that satisfies the constraint.

C. Variations in Non-Markovian Constraints

As a non-Markovian constraint, we consider the following three types.

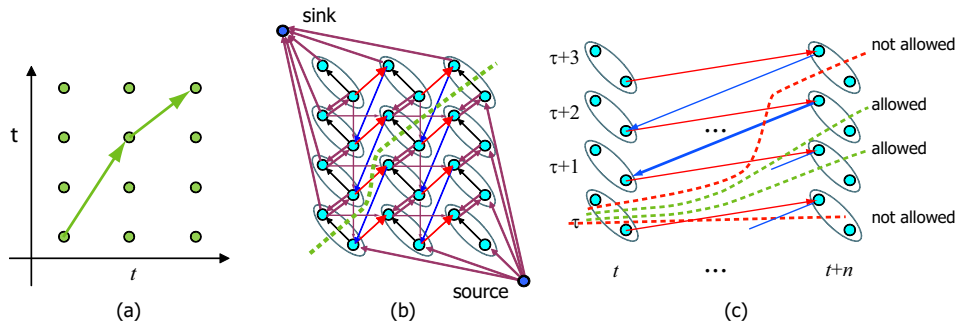

Fig. 3. (a) A warping path on the t–τ grid plane. (b) A graph cut problem equivalent to DTW. The black edge in the graph has a finite cost $d_t(\tau)$, while all other edges have an infinite cost. The red and the blue edges represents the slope constraint with $\alpha_1 = 0$ and $\beta_1 = 2$. (c) Edges (red and blue arrows) for a non-Markovian constraint.

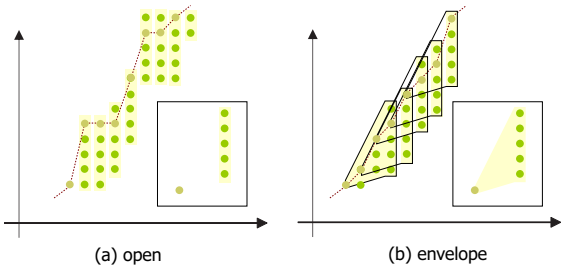

Fig. 4. Difference between open and envelope constraints. Although their parameters are equally $(\alpha_3, \beta_3) = (1, 5)$, their effects are different.

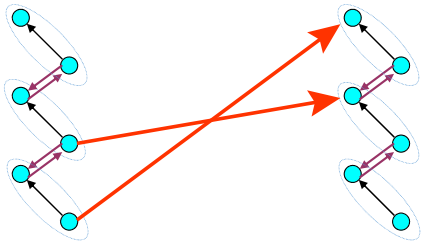

Fig. 5. Collision of multiple non-Markovian constraints. Note that the same constraints exist uniformly over t–τ grid plane.

- *Open constraints*: Fig. 2 (b) shows an open constraint, which is the direct realization of the non-Markovian constraint (5). That is, the distant dependency between t and $t + n$ is regulated. The open constraint is specified by three parameters, n and (α_n, β_n). Note that the open constraint is reduced to a Markovian constraint if $n = 1$.
- *Envelope constraints*: Fig. 2 (c) shows an envelope constraint. The envelope constraint is also specified by (α_n, β_n) but different from the open constraint at the point that the intermediate area between t and $t + n$ is constrained by a linear interpolation of the open constraint. Thus, the warping path cannot deviate from the triangular area of Fig. 2 (c). It is noteworthy that the open and the envelope constraints are different even if their parameters are the same. Fig. 4 illustrates the difference. Because of the interpolated constraints between t and $t+n$, the envelope constraint is tighter and thus the warping path will show less discontinuity than the open constraint.
- *Composite constraints*: Fig. 2 (d) shows a composite constraint. The composite constraint is a combination of multiple open and envelope constraints. In this paper, we focus on the composite constraint only by open constraints.

IV. Learning Constraints

A. Learning An Open Or Envelope Constraint

As noted in Section I, learning the non-Markovian constraint is still an open-problem and thus there is no traditional method for it. We, however, want to observe the optimal non-Markovian constraint of this problem because it will be worthy to know the non-Markovian constraint is really important or not. Since Markovian constraint is a special case of non-Markovian constraints, we can conclude that non-Markovian constraints are useless if a Markovian constraint is selected as the globally optimal constraint.

For this strong demand to the globally optimal constraint, we use the brute-force optimization method. This is the simplest optimization method to try each possible non-Markovian constraint (open or envelope constraint) by changing the combination of n, α_n, and β_n and check its performance by measuring recognition accuracy under the constraint. Then, the non-Markovian constraint with the best recognition accuracy is determined as the best non-Markovian constraint. Clearly, this naive optimization method will provide the globally optimal constraint (although it requires large computations for its repetitive trial process). Note that in the later experiment, the range of n is limited as $n \in [1, 2, \ldots, 10]$ due to the limit of computational resources.

B. Learning A Composite Constraint

For the composite constraint, the above brute-force method becomes computationally intractable. This is because there are a huge number of all possible combinations of the open constraints. Specifically, if we have M possible open constraints and optimize K-composite constraints, we need to repeat $O(M^K)$ recognition experiments. This is intractable even with small K, since M is already a large number by itself.

Consequently, we must give up to obtain the globally optimal composite constraint, and thus resort to greedy-type methods, called *Sequential Forward Selection*(SFS) and *Sequential Floating Forward Selection*(SFFS) [9]. SFS is a simple greedy-type method; it starts with the optimization

of a single constraint. The optimal single constraint found by the brute-force method of IV-A is selected as the first constraint. Then, the optimal second constraint is found by the same brute-force method, while using the first constraint as the fixed constraint. The best second constraint and the first constraint are combined and treated as the 2-composite constraint. This "greedy addition" of k-th constraint to $(k-1)$-composite constraint is repeated until no further improvement. For having a K-composite constraint, we need $O(MK)$ recognition experiments. Although it requires a huge amount of computations, it is far less than $O(M^K)$ for the globally optimal composite constraint and thus efficient.

SFFS is a modified version of SFS. SFFS allows the deletion of a past constraint in the K-composite constraint, as well as the addition of a new constraint. This is a reviewing process of the past greedy additions and thus has a hope of better constraints than those by SFS. SFFS, however, is still a kind of greedy-type method and thus does not guarantee any improvement than SFS. In fact, our experiment (explained in the later section) proved that SFFS provided the same composite constraint as SFS because recognition performance was never improved by the deletion.

An important note for the composite constraint is that there are constraints that cannot coexist. In other words, there is a case that two (or more) constraints cause a collision and do not allow any warping path (that is, any cut cost becomes infinite). Fig. 5 shows an example of collision. Two red arrows suggest two α_ns of two single non-Markovian constraints. If we use them for a composite constraint, no warping path with a finite cost is available. We, therefore, cannot add a new constraint as the k-th constraint to the current $(k-1)$-composite constraint if they cause a collision.

V. Experiment

A. Dataset

For learning non-Markovian constraints and verifying the performance under the learned constraints, handwriting digit patterns from Unipen dataset [10] has been used. The entire Unipen dataset contains about 15,000 patterns Unipen for 10 digit classes (i.e., about 1,500 patterns for each class). Among them, 1,500 patterns were used as training patterns for learning constraints and the remaining 13,500 patterns were used as test patterns. From the training patterns, 5 reference patterns (Y) were generated for each class by using the k-means clustering method.

B. Learned Constraints

The globally optimal open constraint and envelope constraint given by the brute-force method were $(\alpha_8, \beta_8) = (5, 14)$ and $(\alpha_{10}, \beta_{10}) = (1, 19)$, respectively. The composite constraint by SFS was comprised of two open constraints, $(\alpha_4, \beta_4) = (3, 5)$ and $(\alpha_8, \beta_8) = (5, 14)$. Note that the result of SFFS was the same as SFS, as noted in IV-B. Those constraints are illustrated in Fig. 6.

These learning results show the following facts.

- This optimal open constraint was determined between t and $t + 8$, i.e., long distant points. This proves the non-Markovian constraint is better than the conventional Markovian constraint and thus non-Markovian DTW is meaningful.
- This optimal open constraint allows a discontinuous warping path (i.e., a warping path with a sudden increase of τ), like the case of Fig. 4(a).
- Since the parameter α_8 of the optimal open constraint is not zero, thus a path with a long zero-slope part (i.e., a long horizontal part) is not allowed. This will suppress an unnatural DTW. Note that the conventional Markovian constraint $(\alpha_1, \beta_1) = (0, 2)$ allows a long zero-slope part.
- This optimal envelope constraint was almost equivalent to the Markovian constraint $(\alpha_1, \beta_1) = (0, 2)$, in practice. This is because the minimum slope of the envelop constraint is $1/10 = 0.1 \sim 0$ and the maximum is $19/10 = 1.9 \sim 2$.
- The composite constraint was comprised of the optimal open constant of Fig. 6(a) and a shorter-distant open constraint. This shorter-distant open constraint $(\alpha_4, \beta_4) = (3, 5)$ will suppress the discontinuities allowed by $(\alpha_8, \beta_8) = (5, 14)$. Consequently, the composite constraint seems a good compromise between flexibility and tightness for the warping path shape.

C. Recognition Accuracy

Table I shows the recognition accuracy of the test patterns under the learned constraints as well as the conventional Markovian constraint $(\alpha_1, \beta_1) = (0, 2)$. The best accuracy was achieved by the composite constraint. As expected, the envelope constraint and the Markovian constraint provided almost the same accuracy. Although the open constraint can provide more discontinuous warping paths than the Markovian, its positive and negative effects were canceled and thus its accuracy was almost the same as the Markovian.

D. A Closer Look of Matching Results

Fig. 7 (a) shows the case with the improvement by the composite constraint from the Markovian constraint. A careful observation of the warping paths of those cases reveals that *the composite constraint does allow small and local discontinuities but does not allow the large global warping.* The latter point can be confirmed by more diagonal warping paths by the composite constraints. Consequently, non-Markovian DTW with the composite constraint can realize a different property in its local and global warping range. Especially, the global regulation is realized by the effect of the long-distant constraint.

VI. Conclusion

This paper tried to extend the horizon of DTW by introducing various non-Markovian constraints, which was realized by using graph cut solution instead of dynamic programming solution. Our main contributions were as follows: (i) The optimal DTW constraint for the handwriting recognition was a non-Markovian constraint and thus the conventional Markovian DTW has a limitation in its performance. (ii) Using non-Markovian constraints, we can control not only local but also

TABLE I. RECOGNITION ACCURACY (%) UNDER THE LEARNED CONSTRAINTS.

Constraint type	Class 0	1	2	3	4	5	6	7	8	9	total
Open $(\alpha_8, \beta_8) = (5, 14)$	87.457	81.567	94.956	96.842	81.950	83.445	96.190	91.424	90.589	90.466	89.489
Envelope $(\alpha_{10}, \beta_{10}) = (1, 19)$	86.715	82.938	94.331	96.178	82.559	85.052	96.264	92.899	89.386	88.264	89.459
Composite $(\alpha_4, \beta_4) = (3, 5)$& $(\alpha_8, \beta_8) = (5, 14)$	85.663	85.333	95.520	95.886	85.430	84.278	96.134	94.000	90.288	90.975	90.351
Conventional (Markovian, $(\alpha_1, \beta_1) = (0, 2)$)	86.741	82.857	94.229	96.187	82.472	85.134	96.292	92.644	89.231	88.155	89.394

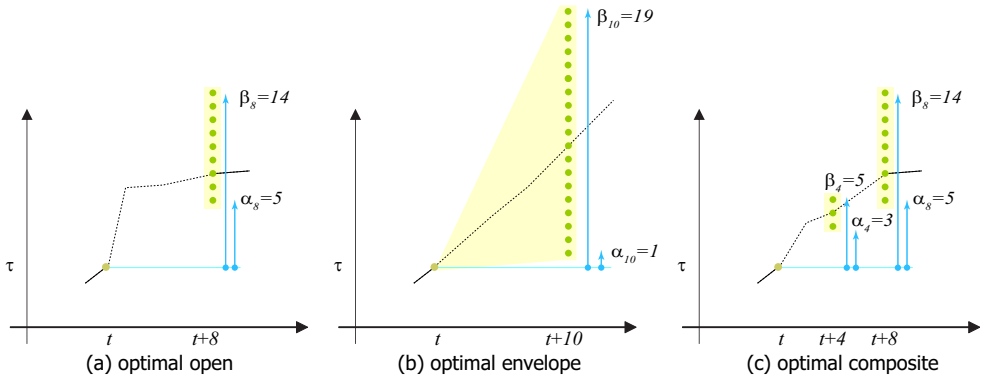

Fig. 6. Learned constraints.

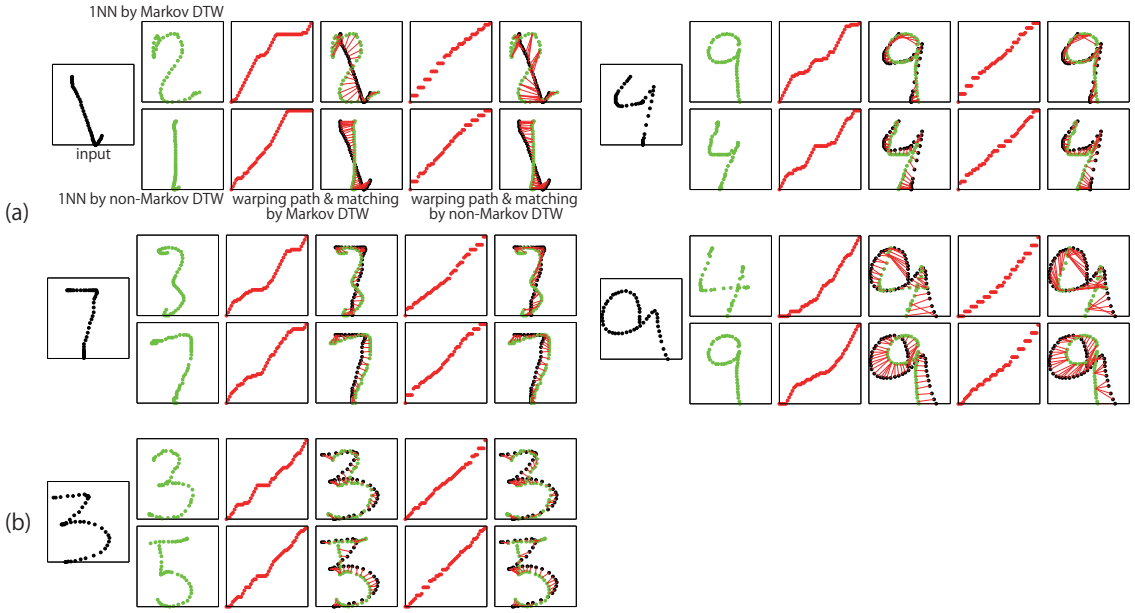

Fig. 7. DTW examples (a) improved and (b) degraded by the learned non-Markovian constraint.

global shape of the warping path. Future work will focus on various applications of the non-Markovian DTW and discovery of the optimal constraints at individual applications.

REFERENCES

[1] P. W. Felzenszwalb and R. ZabihC "Dynamic Programming and Graph Algorithms in Computer Vision," *TPAMI*, 33(4), 2011.

[2] H. Ishikawa, D. GeigerC "Occlusions, Discontinuities, and Graph Algorithms in Computer Vision," *ECCV*, 1998.

[3] S. Uchida, M. Fukutomi, K. Ogawara, and Y. Feng, "Non-Markovian Dynamic Time Warping", *ICPR*, 2012.

[4] V. Frinken, R. Kakisako and S. Uchida, "A Novel HMM Decoding Algorithm Permitting Long-Term Dependencies and its Application to Handwritten Word Recognition", *ICFHR*, 2014.

[5] D. Cremers and L. Grady, "Statistical Priors for Efficient Combinatorial Optimization via Graph Cuts", *ECCV*, 2006.

[6] P. Krähenbühl and V. Koltun, "Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials", *NIPS*, 2011

[7] N. D. F. Campbell, K. Subr, and J. Kautz, "Fully-Connected CRFs with Non-Parametric Pairwise Potentials", *CVPR*, 2013.

[8] K. Ogawara, M. Fukutomi, S. Uchida, and Y. Feng, "A Voting-Based Sequential Pattern Recognition Method", *PLoS ONE*, 8(10), 2013.

[9] P. Pudil, J. NovovičováCand J. Kittler, "Floating Search Methods in Feature Selection," *PRL*, 15(11), 1994.

[10] E. H. Ratzlaff, "Methods, Report and Survey for the Comparison of Diverse Isolated Character Recognition Results on the UNIPEN Database", *ICDAR*, 2003.