Integrated Event Recognition from Multiple Sources

Hiroaki KAWASHIMA Takashi MATSUYAMA

Graduate School of Informatics, Kyoto University
Yoshida-Honmachi, Sakyo, Kyoto 6068262, JAPAN
e-mail: {hiroaki, tm}@vision.kuee.kyoto-u.ac.jp

Abstract

This paper proposes a system architecture for event recognition that integrates information from multiple sources (e.g., gesture and speech recognition from distributed sensors in the real world). The proposed system consists of multiple recognizers named Continuous State Machines (CSMs). Each CSM has a state transition rule in a continuous state space and classifies time-varying patterns from a single source. Since the rule is defined as a simplification of Kalman filter (i.e., the next state is deduced from the trade-off scheme between input data and model's prediction), CSMs support dynamic time warping and robustness against noise. We then introduce an interaction method among CSMs to classify events from multiple sources. A continuous state space (i.e., vector space) allows us to design interaction as recursively minimizing an energy function. This interaction enables the system to dynamically focus over the multiple sources, and improves reliability and accuracy of classifying events in dynamically changing situations (e.g., the object is temporally occluded from one of multiple cameras in a gesture recognition task). Experimental results on gesture recognition by two cameras show the effectiveness of our proposed system.

1. Introduction

Event recognition such as gesture or speech recognition in the real world is one of the tough problems. This is because we cannot expect the system to obtain complete, reliable information in dynamically changing situations. Most of the reported recognition systems, therefore, heavily rely on the specific constraints (e.g., objects has no occlusions in gesture recognition tasks [1, 2]). To design a system that recognizes events robustly in the unconstrained environment, we believe that the system need to integrate information from distributed multiple sensors.

This paper proposes a system architecture for event recognition that integrates information from multiple sources. The proposed system consists of multiple recognizers, where each recognizer obtains input data from a single source (Fig.1). The key issue to design this kind of modular architecture rests in (1) what model to choose for the recognizers and (2) how to integrate all the recognizers.

Firstly, we propose a state space model named *Continuous State Machines* (CSMs) for recognizers. CSMs are designed based on the context of using an energy function. This is because an energy function can deal with real values and is easy to interpret the meaning of parameters. An existing continuous state based sequential pattern recognition model such as the Recurrent Neural Network[3], on the other hand, deals with only sequential binary patterns.

Secondly, we introduce a dynamic interaction mechanism among CSMs based on co-occurrence of their states that is learned before the recognition phase. Since a continuous state space has a metric, designing an interaction with another energy function is straightforward.

Introducing the interaction among CSMs allows the system to integrate multi-lateral and multi-modal information not at a time but dynamically (i.e., changing focus on specific sources and making up for temporal lack of information with each other). We can therefore expect that the system works robustly in unconstrained situations, where some of the sources temporally have inadequate information to classify the events.

2. System Architecture

The system that we propose consists of M-multiple CSMs as shown in Fig.1, where each CSM module observes a time-varying pattern from a single source. Note that the number of CSMs and that of sources are the same in our system. We call this system an Integrated CSM (ICSM).

Continuous State Machines: To naturally classify time-variant patterns (e.g., to support dynamic time warping and noise robustness), we propose a model that uses state transition in a continuous state space. Which is deduced from trade-off scheme of input data and model's prediction.

As shown in Fig.2, time-variant patterns are embedded in a state space as their corresponding trajectories by a simple batch algorithm in the learning phase. In the recognition phase, on the other hand, a state of each CSM at an instant

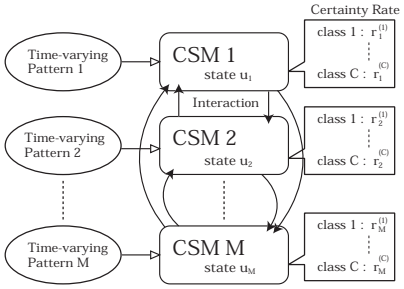


Figure 1: System Architecture

is represented by a point in its state space. When an unclassified time-variant pattern is observed, the CSM's state thus changes along one of the learned trajectories. As a result, we can classify the time-variant pattern depending on the trajectory by which the state is attracted. The design of the state space and trajectories are described in detail in section 3.3.

A continuous state space has the following advantages against a discrete state space[1, 2, 4]. (1) We need not to segment a dynamic event into finite states and optimize the number of the states. This facilitates a more natural representation of events. (2) The distance between two states can be simply defined as a norm of a vector whose terminals correspond to the two states. Accordingly, the interaction can be designed as recursively minimizing an energy function. This interaction integrates all the estimated states from the other CSMs.

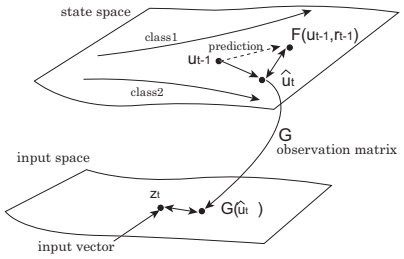


Figure 2: Trajectories of two classes in the state space

Certainty Distribution: Since observable patterns are time-variant, confidence of each CSM dynamically changes. To measure this confidence, we define *certainty* distribution for each CSM. Suppose that CSM m has certainty $r_{m,t}^{(c)}$ for class $c\,(c = 1, ..., C)$ at time t, the certainty distribution for m is represented as vector $\boldsymbol{r}_{m,t} = (r_{m,t}^{(1)}, ..., r_{m,t}^{(C)})^T$ (superscript T denotes the transpose). The dynamics of individual CSMs and interaction among CSMs are controlled by this certainty distribution during the recognition phase.

Interaction among CSMs: Once all CSMs update their own states, each CSM estimates the states of the other CSMs. Each CSM then integrates all the estimated states with their own state. In this paper, we design this integration from the point of dynamic focusing on specific sources. Thus, the influence of the CSM is temporally set to be lower than the other CSMs, when its input has inadequate information for classification.

3. Dynamics of a Single CSM

3.1. State Transition

Assume that all the input vectors $\boldsymbol{z}_t$ are independent and that the state at time t satisfies the Markov property. The next state of the CSM is thus deduced from both top-down information of a prediction and bottom-up data of an external input. The dynamics of the CSM is therefore defined as minimizing the following energy function at each time t:

$$E_1(\boldsymbol{u}_t) = (1-\kappa)||\boldsymbol{u}_t - \mathcal{F}(\boldsymbol{u}_{t-1}, \boldsymbol{r}_{t-1})||^2 + \kappa||G\boldsymbol{u}_t - \boldsymbol{z}_t||^2, \quad (1)$$

where $||\cdot||$ denotes the Euclidean norm of a vector. $\boldsymbol{u}_t, \boldsymbol{z}_t$ and $\boldsymbol{r}_t$ denote a state, an input and certainty distribution at time t, respectively. The first term of the right-hand side of Eq.(1) denotes prediction error in the state space, and the second term denotes the observation error in the input space. $\mathcal{F}(\cdot)$ is a prediction function that predicts the next state of CSM and generally it is non-linear. G is an observation matrix that maps the state vector into the input space. The weight κ $(0 \leq \kappa \leq 1)$ determines the balance between predictions and inputs. This can be regarded as a trade-off scheme such as Kalman filters [5]. The state transition rule follows from Eq.(1):

$$\hat{\boldsymbol{u}}_t = \{(1-\kappa)I + \kappa G^T G\}^{-1} \times \{(1-\kappa)\mathcal{F}(\boldsymbol{u}_{t-1}, \boldsymbol{r}_{t-1}) + \kappa G^T \boldsymbol{z}_t\}, \quad (2)$$

where I is an identity matrix. We call $\hat{\boldsymbol{u}}_t$ a *hypothetical state* at time t. This is because $\hat{\boldsymbol{u}}_t$ is a temporal state, i.e., it is updated by the interaction among CSMs (section 4).

3.2. Prediction based on Certainty

To predict the state at time t from $\boldsymbol{u}_{t-1}$, we employ the following function:

$$\mathcal{F}(\boldsymbol{u}_{t-1}, \boldsymbol{r}_{t-1}) = \sum_{c=1}^{C} r_{t-1}^{(c)} \mathcal{F}^{(c)}(\boldsymbol{u}_{t-1}), \quad (3)$$

where $\mathcal{F}^{(c)}(\boldsymbol{u}_{t-1})$ is a conditional prediction function given below. This equation indicates that prediction function $\mathcal{F}(\cdot)$ is approximated as a linear combination of conditional prediction functions of each class c, where the coefficients of the linear combination are certainty $r_{t-1}^{(c)}$ at time $t-1$.

For simplification, we assume that conditional prediction function $\mathcal{F}^{(c)}(\boldsymbol{u}_{t-1})$ is approximated as a linear function of

the entries of $\boldsymbol{u}_{t-1}$:

$$\mathcal{F}^{(c)}(\boldsymbol{u}_{t-1}) = W^{(c)}\boldsymbol{u}_{t-1}, \tag{4}$$

where $W^{(c)}$ is a transition matrix of class c which is learned in learning phase.

In addition, we define certainty by:

$$r_t^{(c)} = k_t \exp(-d_t^{(c)}/\sigma^2), \tag{5}$$

where $d_t^{(c)}$ is the squared error between a prediction and an input data in the input space:

$$d_t^{(c)} = ||\boldsymbol{z}_t - G\mathcal{F}^{(c)}(\boldsymbol{u}_{t-1})||^2. \tag{6}$$

We remark that $r_t^{(c)}$ $(c = 1, ..., C)$ are normalized to satisfy $\sum_{c=1}^{C} r_t^{(c)} = 1$. *Softmax function* is used in Eq.(5) to make the range of $r_t^{(c)}(c = 1, ..., C)$ be $[0, 1)$.

3.3. Design of the State Space

The state space is designed to discriminate classes of observed time-varying patterns. Since trajectories of time-varying patterns in the input space may have intersection or overlapping within one pattern or with other patterns, we extend the dimension of the input space and generate a state space to separate corresponding trajectories. In other words, the state space is designed as the direct-product of the input space and the discrimination space.

Let $S_I^{(1)}, ..., S_I^{(C)}$ be C time-varying patterns to be learned, where matrix $S_I^{(c)} = [\boldsymbol{s}_{I,1}^{(c)}, ..., \boldsymbol{s}_{I,L-1}^{(c)}, \boldsymbol{s}_{I,L}^{(c)}]$ denotes a trajectory that corresponds to class c, and vector $\boldsymbol{s}_{I,t}^{(c)}$ denotes a point in the input space. An additional sequential pattern $S_O^{(c)} = [\boldsymbol{s}_{O,1}^{(c)}, ..., \boldsymbol{s}_{O,L-1}^{(c)}, \boldsymbol{s}_{O,L}^{(c)}]$ is then appended to $S_I^{(c)}$ to obtain $S^{(c)} = [S_I^{(c)T}, S_O^{(c)T}]^T$. The dimension of $\boldsymbol{s}_{I,t}^{(c)}$, $\boldsymbol{s}_{O,t}^{(c)}$ and $\boldsymbol{s}^{(c)} = [\boldsymbol{s}_{I,t}^{(c)T}, \boldsymbol{s}_{O,t}^{(c)T}]^T$ are N_I, N_O and $N(= N_I + N_O)$, respectively.

One of the straightforward codings of $S_O^{(c)}$ to discriminate all the time-varying patterns is as follows. (1) Choose a static initial pattern $\boldsymbol{o}$ and goal patterns $\boldsymbol{g}^{(1)}, ..., \boldsymbol{g}^{(C)}$. The initial pattern is common in all the classes, and the goal patterns are mutually orthogonal. Each goal pattern $\boldsymbol{g}^{(c)}$ is also orthogonal to $\boldsymbol{o}$ and all the norms of these vectors are set to be the same. (2) Monotonously interpolate $\boldsymbol{o}$ and $\boldsymbol{g}^{(c)}$ to generate $S_O^{(c)}$.

Adding additional sequential patterns to all the time-varying patterns allows the corresponding trajectories of the class to be gradually separated from each other in the state space (Fig.2).

As a result of the state space designed above, observation matrix becomes $G = [I_{N_I}|O]$. We call the extended part of the state *output state* $\boldsymbol{u}_{O,t} = [O|I_{N_O}]\boldsymbol{u}_t$, since the states of events are essentially represented by the output states.

Batch Learning of CSMs: Weight matrix $W^{(c)}$ in Eq.(4), which predicts $\boldsymbol{s}_{t+1}^{(c)}$ from $\boldsymbol{s}_t^{(c)}$, satisfies

$$S'^{(c)} = W^{(c)}S^{(c)}, \tag{7}$$

where $S'^{(c)} = [\boldsymbol{s}_2^{(c)}, ..., \boldsymbol{s}_L^{(c)}, \boldsymbol{s}_L^{(c)}]$. Note that we assume that a prediction of the last state is itself (i.e., $\boldsymbol{s}_{L+1}^{(c)} = \boldsymbol{s}_L^{(c)}$). Eq.(7) gives a transition matrix for class c:

$$W^{(c)} = S'^{(c)}S^{(c)+}, \tag{8}$$

where $S^{(c)+}$ is the generalized inverse matrix of $S^{(c)}$.

3.4. Decision process

We introduce the following two decision processes which are compared in the experiments in section 5.

(A) Similarity: The state of a CSM is attracted by the nearest trajectory in the state space during the recognition phase. We, therefore, evaluate the class of observed patterns by the similarity of the output states between the goal patterns of each class. Here we use the normalized correlation as the similarity:

$$e_t^{(c)} = \frac{\boldsymbol{g}^{(c)}}{||\boldsymbol{g}^{(c)}||} \cdot \frac{\boldsymbol{u}_{O,t}}{||\boldsymbol{u}_{O,t}||}, \tag{9}$$

where $e_t^{(c)}$ takes the range of $[-1, 1]$. Since goal patterns are designed to be orthogonal with each other, the similarity of two classes does not reach 1 simultaneously. Accordingly, a classification result is obtained by $\arg\max_{(c,t)} e_t^{(c)}$ $(t = t_{\rm start}, ..., t_{\rm end})$, where we assume that time-variant patterns are segmented beforehand in the range of $[t_{\rm start}, t_{\rm end}]$.

(B) Accumulated Prediction Errors: The errors between prediction and input vectors are calculated by Eq.(6) at each transition. We, therefore, evaluate the class by the accumulation of the prediction errors:

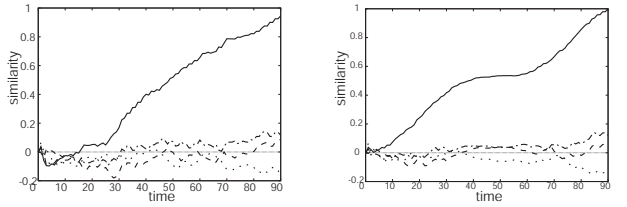
$$D_t^{(c)} = \sum_{t'=t_{\rm start}}^{t} d_{t'}^{(c)}. \tag{10}$$

Accordingly, the result class is obtained by $\arg\min_c D_{t_{\rm end}}^{(c)}$.

3.5. Evaluations of a Single CSM

We randomly generated four binary patterns A, B, C, D, and then monotonously interpolated them as four time-variant patterns ABC, ABD, DAC, DBC. We also trained a CSM by these four sequences. Next, we input one of the patterns to the CSM with noise or with non-linear temporal fluctuation.

Figure 3 shows the similarity of all classes in the recognition phase. We see that the corresponding similarity to the input class reaches near 1 in spite of noise and temporal fluctuation.

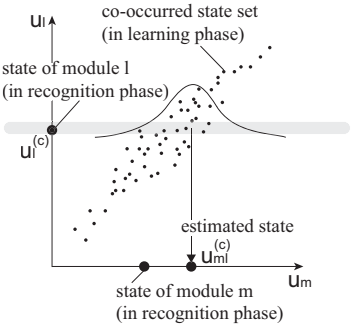


(a) 20% noise (b) non-linear temporal fluctuation

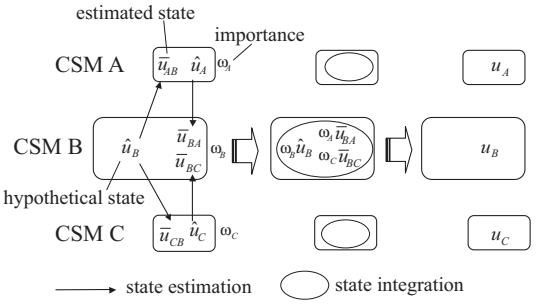
Figure 3: Robustness against noise and time warping

4. Interaction among CSMs

The interaction among CSMs is designed based on the two following functions (Fig.4). (1) Estimation of the states of other $M-1$ CSMs from its own state, which is based on co-occurrence of states. This estimation makes up for temporal lack of information with each other. (2) Integration of the estimated states at each CSM with dynamic weighting. This integration enables ICSMs to dynamically focus on more reliable information for event recognition.



(a) state estimation



(b) process of interaction

Figure 4: Interaction among CSMs

4.1. Learning Co-occurrence of States

Since we designed state transition in not a discrete but a continuous state space, we cannot represent the co-occurrence of states as a matrix of probability. The most straightforward way to represent the co-occurrence is store the co-occurred state vectors as a multi-dimensional histogram. For example, to learn the co-occurrence between states $\boldsymbol{u}_{m,t}$ and $\boldsymbol{u}_{l,t}$ in the output space, we consider histogram $\mathcal{H}(\boldsymbol{u}_{O,m,t}, \boldsymbol{u}_{O,l,t}|c)$. Here the output states $\boldsymbol{u}_{O,m,t}$ is used in stead of the state vector $\boldsymbol{u}_{m,t}$. This is because output states represent more abstracted information of the observed events.

In fact, kd-tree[6] is used to store sampled vectors. We then search data vectors with respect to a key vector during the recognition phase.

4.2. Estimation of States

To estimate a state of CSM m based on a state of CSM l, we assume that the conditional probability density function of the state $\boldsymbol{u}_m$ is given as the Gaussian distribution: $p(\boldsymbol{u}_{O,m,t}|\boldsymbol{u}_{O,l,t} = \hat{\boldsymbol{u}}_{O,l,t}, c) = N(\bar{\boldsymbol{u}}^{(c)}_{O,ml,t}, \Sigma^{(c)}_{O,ml,t})$ where $\bar{\boldsymbol{u}}^{(c)}_{O,ml,t}$ is the mean vector and $\Sigma^{(c)}_{ml,t}$ is the covariance matrix of the distribution. This distribution is obtained from the co-occurrence histogram $\mathcal{H}(\boldsymbol{u}_{O,m,t}, \boldsymbol{u}_{O,l,t}|c)$ by searching the nearest q points with respect to $\hat{\boldsymbol{u}}_{O,l,t}$ (Fig.4 (a)).

Using the linear combination of the estimated mean states whose weights are certainty for each class allows us to obtain a estimated state of CSM m from CSM l:

$$\bar{\boldsymbol{u}}_{O,ml,t} = \sum_{c=1}^{C} r^{(c)}_{l,t} \bar{\boldsymbol{u}}^{(c)}_{O,ml,t} \ . \tag{11}$$

4.3. Dynamic Control of Importance

Since observable information is dynamically changing, we have to dynamically control the system so that it focuses on reliable data and ignores unreliable data. We, therefore, we define *importance* for each CSM as a confidence measure. In this paper, we define importance based on the difference between the two most competitive prediction errors $d^{(c)}_{m,t}$ (Eq.(6)):

$$\omega_{m,t} = (d^{\mathrm{2nd}}_{m,t} - d^{\mathrm{min}}_{m,t}) / \sum_{c=1}^{C} d^{(c)}_{m,t}, \tag{12}$$

where $d^{\mathrm{min}}_{m,t}$ and $d^{\mathrm{2nd}}_{m,t}$ denote the first and second minimum squared prediction error, respectively.

4.4. Integration of Estimated States

Integration of the own hypothetical state with the estimated states from the other CSMs is defined by minimizing the following evaluation function at each time t:

$$E_2(\boldsymbol{u}_{m,t}) = (1-\gamma)\omega_{m,t}||\boldsymbol{u}_{m,t} - \hat{\boldsymbol{u}}_{m,t}||^2 + \gamma \sum_{l=1, l\neq m}^{M} \omega_{l,t}||\boldsymbol{u}_{O,m,t} - \bar{\boldsymbol{u}}_{O,ml,t}||^2, \tag{13}$$

where γ $(0 \leq \gamma \leq 1)$ denotes *dependence* on the other CSMs. The first term of the right-hand side denotes the squared error from its own hypothetical state and the second term denotes the sum of the squared errors from the estimation of the other CSMs, where each error is weighted by importance defined in Eq.(12).

5. JSL Recognition by an ICSM

We chose seven words from Japanese Sign Language (JSL), where all the words consist of stylized upper-body gestures. 21 samples by three people for each gesture were synchronously captured from front and side cameras. 24×24 down-sampled silhouette images of each frame were used as input feature vectors for two CSMs. The dimension of each CSM was 1224 $(N_I = 576, N_O = 200)$.

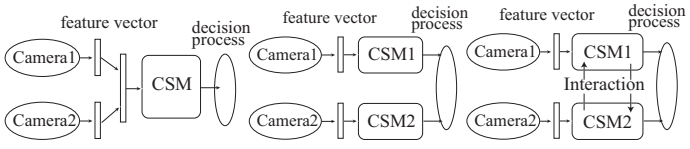
In the learning phase, we chose 12 samples from each gesture for training. Then, in the recognition phase, the other samples were used for testing. We changed the combination of training samples in 10 ways and calculated the recognition rates from 630 trials.

Table 1 shows the recognition rates of the two independent CSMs and the following three architectures (Fig.5).

(a) Feature-level Integration (FI) : A single CSM is used (Fig.5(a)). The input data of the CSM is a connected vector of two feature vectors obtained from front and side cameras at each frame. The dimension of the CSM was 2448 $(N_I = 1152, N_O = 400)$.

(b) Decision-level Integration (DI) : Two independent CSMs are used (Fig.5(b)). Each input data is a feature vector of each camera. Then the result is integrated in the decision-level as the average of the similarity or the accumulated prediction errors (section 3.4).

(c) State-level Integration (SI) : An ICSM is used (Fig.5(c)). This architecture is considered as DI with dynamic interaction among the state space of CSMs.



(a) Feature-level Integration (FI) (b) Decision-level Integration (DI) (c) State-level Integration (SI)

Figure 5: Three architectures for comparison

As shown in Table 1, the recognition rates of CSM2 (for the side camera) is much lower than that of CSM1 (for the front camera). This is because gestures of side view are very similar among some classes. While the rates of FI and DI are between those of CSM1 and CSM2, SI takes the highest rate among all the other architectures include CSM1, especially using decision process (A). We remark that SI dynamically focused on one of the consisting CSMs and temporally inhibited the other CSM whose input data were unreliable source. On the other hand, the other architectures were totally drawn by unreliable input data.

Comparing the two decision processes, (B) takes lower recognition rates except DI. This is because (B) is more sensitive to noise compared to (A), since (B) evaluates errors in the input space.

Table 1: Recognition rates by two decision processes(%) See section 3.4 for decision process (A) and (B). (The numbers in the parenthesis denote the recognition rates of the ICSM without dynamic control of importance.)

	CSM1	CSM2	FI	DI	SI
(A) Max.Sim.	86	53	76	80	88 (76)
(B) Min.Err.	85	51	73	83	82 (81)

6. Conclusion

This paper presented a system architecture called ICSM for integrated event recognition from multiple sources.

Introducing two energy functions that derive inner dynamics and inter dynamics of CSMs in a continuous state space, an ICSM can change focus on more important sources for event recognition with robustness against noise and temporal fluctuation.

References

[1] J. Yamato, J. Ohya and K. Ishii: "Recognizing human action in time-sequential images using hidden markov model", *Proc. CVPR'92*, pp. 379–385 (1992).

[2] M. Brand and N. Oliver: "Coupled hidden markov models for complex action recognition", *MIT Media Lab Vision and Modeling TR407* (1997).

[3] M. Morita: "Memory and learning of sequential patterns by nonmonotone neural networks", *Neural Networks*, **9**, 8, pp. 1477–1489 (1996).

[4] A. Bobick and A. Wilson: "State-based approach to the representation and recognition of gesture", *IEEE Trans. PAMI*, **19**, 12, pp. 1325–1337 (1997).

[5] H. Segawa, H. Shioya, N. Hiraki and T. Totsuka: "Constraint-conscious smoothing framework for the recovery of 3d articulated motion from image sequences", *Proc. FG 2000*, pp. 476–481 (2000).

[6] J. Bentley: "Multidimensional binary search trees used for associative searching", *Commun. ACM*, **18**, 9, pp. 509–517 (1975).