Internal-to-internal transition method for consecutive hierarchical template matching

Ho Gi Jung

Department of Automotive Engineering, Hanyang University, 222 Wangsimni-ro, Seongdong-gu, Seoul, 133-791, Republic of Korea
E-mail: hogijung@hanyang.ac.kr

Abstract: This study proposes a method reducing the tree traversal cost by first investigating the most probable node instead of the root node when a hierarchical template matching is consecutively applied to the object. In particular, this study gives a novel viewpoint that a consecutive hierarchical template matching could be regarded as a transition of a hierarchical finite state machine, and then it proposes a novel method considering the transitions between the internal nodes of the hierarchical template tree. The proposed method is verified by applying it to pedestrian silhouette detection.

1 Introduction

Template matching is one of the tools that have been used for object detection and classification for a long time in the field of computer vision. If the templates are acquired from object samples to be detected, the system can be constructed easily. Such an approach is referred to as exemplar-based. If the number of templates increases to be impractical for one-by-one comparison, as when the target object includes various kinds or has a wide range of deformation, hierarchical template matching using a hierarchical template tree is frequently adopted. Template matching using a hierarchical template tree is reported to be successful in the application of pedestrian detection with various appearance changes [1], human hand detection and pose estimation [2] and object recognition with various targets. In particular, Gavrila [1] defines the dissimilarity measurements between two pedestrian silhouettes with chamfer matching error, and proposes a method constructing a hierarchical template tree of pedestrian silhouettes by recursively applying a $k$-medoid clustering algorithm from the bottom level.

Meanwhile, when the appearance or pose continuously changes like a pedestrian or a human hand, the change can be modelled as a Markov process in the parameter space [3–5]. This means that detecting an object or estimating its pose at a certain moment can be implemented by estimating parameters that maximise a posteriori probability considering the previous detection or estimation results. In particular, Rogez et al. [3] represent the parameters of a walking pedestrian in a two-dimensional (2D) manifold, where the action is represented by a 1D manifold and the viewpoint by another 1D manifold. As the viewpoint parameter is intrinsically cyclic, the viewpoint axis is represented by a circle and as human walking action is also cyclic; the action axis is also represented by a circle. Consequently, these two axes are incorporated into a 2D manifold, which lies on a closed cylinder topologically equivalent to a torus.

A temporal dynamic model can be represented by a hierarchical tree structure, and the robustness and operation speed of object detection and pose estimation can be improved by a coarse-to-find approach [2, 6, 7]. In [2], Stenger et al. propose a hand pose tracking combining a hierarchical template tree and Markov process, which learns the temporal transition probabilities between hand poses using the learning data, and then exploits the previous matching results as well as the current image to detect a human hand and estimate its pose. In [6], Rogez et al. partition the human pose parameter space by the torus manifold discretisation and 2D pose space clustering, and then construct the hierarchical tree structure by merging similar classes in a bottom-up manner as in [1]. This means that the human silhouette can be classified by a hierarchical tree structure and we can estimate the probability that a tree node will be matched at the next time step after a specific tree node is matched, because a tree node is corresponding to a region on the human pose parameter space.

This paper proposes a method reducing the number of comparisons, that is, template matching, when hierarchical template matching-based object detection is applied to consecutive images. The proposed method originates from a question: ‘Even when we can surely predict that a specific node of a hierarchical template tree will be matched in the next time step, should we start the tree traversal from the root node?’ Although previous methods like [2, 6, 7] combining a hierarchical template matching and a temporal integration exploit the previous matching results to estimate the probability that a specific node will be matched in the current time step, they should start from the root node in each time step. In contrast, this paper proposes that if the node that will be matched in the next time step can be surely predicted, the tree traversal can start from the node. If the most probable
node turns out to be improper, that is, ‘not matched,’ the whole template tree is investigated from the root node as in the previous methods. If the prediction about the next time step is correct, the proposed method will require a smaller number of matching compared with the method unconditionally starting from the root node. The proposed method can be thought of as a compromise exploiting the efficiency of Bayesian filtering while avoiding its disadvantage that it can divert away from the correct solution [7].

In the learning phase, the proposed method estimates cost, that is, the number of template matching until reaching the leaf level, of each node if the tree traversal starts from the node after a specific node is matched in the previous time step and then, the node requiring the lowest cost is selected as the traversal starting point of the node matched in the previous time step. In the execution phase, if a leaf node is matched, its traversal starting point will be firstly investigated in the next time step. If the matching error of the node is above a threshold, matching will continue from the root node. In particular, the proposed method exploits the transition between the internal nodes of a hierarchical template tree by regarding two hierarchical template matchings in series as a transition of a hierarchical finite state machine. In this case, as the traversal starting point of a node is not restricted to the leaf level, the first investigation is not just a template matching with the node, but a sub-tree traversal starting from it. As the root node can be the traversal starting point of each node, the proposed method can be thought of as a generalisation of the previous methods. As the proposed method does not construct the parameter space and directly exploits the hierarchical template tree like [7], it does not require the parameter model of the target object unlike [2, 6].

2 Leaf-to-leaf transition method

Assume that we are trying to detect a pedestrian in consecutive images by applying a hierarchical template matching and the hierarchical template tree has L levels. A pedestrian corresponding to the ath leaf node is detected in the time step t and one corresponding to the bth leaf node is detected in the time step t + 1. Fig. 1 depicts the situation. If the state variable of time step t, which can be one of the leaf nodes, is denoted by s′, the situation can be denoted by s′ = n(L, i), s′+1 = n(L, j), where n(L, i) denotes the i th node of the L th level. As L is the lowest level to which leaf nodes belong, n(L, i) is one of the leaf nodes.

Hereafter (i, l) is an index corresponding to the l th node of the L th level and it can uniquely designate any node. Such an expression is referred to as a tree coordinate.

As the silhouettes of a walking pedestrian are temporally related with each other, the sequence of the leaf nodes of a template tree corresponding to consecutive images can be modelled by a Markov process. In other words, the set of the leaf nodes of a template tree becomes the set of Markov process states and the silhouette change of a walking pedestrian becomes a transition between two states. If the first-order Markov (FOM) model is used, the state estimate of time step t + 1, s′+1, can be calculated with the state of current time step, s′ = n(L, i), as

\[ s′+1 = \arg \max_{n(L, j)\in\mathbb{N}_L} p\left(s′+1 = n(L, j) | s′ = n(L, i) \right) \]  

where \( \mathbb{N}_L \) denotes the node set of level L and \( p(s′+1 = n(L, j) | s′ = n(L, i)) \) denotes the probability that n(L, j) will be matched in the next time step after n(L, i) is matched. Consequently, s′ = n(L, i), s′+1 = n(L, j) means that the ath leaf node is matched in the current time step and the bth leaf node is predicted to be matched in the next time step.

As the FOM model-based prediction cannot be always correct, that is, \( \forall n(L, i) \in \mathbb{N}_L, \quad p(s′+1 = n(L, j) | s′ = n(L, i)) = 1 \) cannot be true, two supplementations are required to detect a pedestrian in consecutive images: a method determining whether the prediction is correct or not and a countermeasure for when the prediction is incorrect. To discuss the improvement of an FOM model-based state prediction about the silhouette of a walking pedestrian, the simplest method is used as a basic framework, as below. This method will be referred to as leaf-to-leaf transition method. The procedure determining the next state prediction of each node is conducted in the offline mode during the learning phase. Resultantly, every leaf node has information about which leaf node will be investigated first in the next time step after itself. The following steps 1)–3) are the pseudo code of the leaf-to-leaf transition method, and Fig. 2 shows the concept diagram.

1) The leaf node corresponding to the current image, n(L, i), is detected.
2) In the next time step, it is checked whether there is a pedestrian corresponding to the prediction \( s′+1 = n(L, i) \)
3) If the previous step 2) fails, that is there is no pedestrian corresponding to the prediction, the hierarchical template matching is applied with the root node as the starting point.

![Fig. 1](image1.png) If the ath leaf node is matched in the time step t and the bth leaf node is matched in the time step t + 1, the situation could be modelled as a state transition from a to b.

![Fig. 2](image2.png) Concept diagram of leaf-to-leaf transition method.
The expected template matching number (ETMN) $E_{(l,0)}$ of a node $n_{(l,0)}$ is defined as the expected number of template matching when the hierarchical template matching is applied with the node designated by the tree coordinates $(l,0)$ as the starting point. A leaf node needs only one template matching and a node just above the leaf level needs template matchings as many as its child node number. For the higher level, ETMN can be recursively defined with the ETMNs of child nodes as (see equation at the bottom of the page)

where $C(n_{(l,0)})$ denotes the set of the child node of $n_{(l,0)}$ and $|C(n_{(l,0)})|$ denotes the cardinality of the child node set and $p(n_{(l+1,j)},n_{(l,0)})$ is the probability that $n_{(l+1,j)}$ will be matched as the optimal branch among the child nodes after $n_{(l,0)}$ is matched.

If the temporal relation between consecutive images is not used, the hierarchical template matching should start from the root node all the time. Such a method is referred to as root-to-root transition method. Detecting a pedestrian in consecutive images based on the method needs cost as much as $E_{(1,1)}$.

The cost of the leaf-to-leaf transition method, $C_{\text{leaf-to-leaf}}$, can be calculated as (see (2))

\[
C_{\text{leaf-to-leaf}} = \sum_{n_{(l,0)} \in N_l} p(n_{(l,0)}) \left( \sum_{n_{(l,0)} \in N_l} p(n_{(l,0)}) \right) 
\times \left( p(s^{l+1} = s^{l+1}(n_{(l,0)})) |s' = n_{(l,0)}| \right) 
\times \left( 1 + E_{(1,1)} \right)
\]

\[
\sum_{n_{(l,0)} \in N_l} p(n_{(l,0)}) \left( E_{(1,1)} - p(s^{l+1} = s^{l+1}(n_{(l,0)})) |s' = n_{(l,0)}| \right) 
\times \left( 1 + E_{(1,1)} \right)
\]

\[
\sum_{n_{(l,0)} \in N_l} p(n_{(l,0)}) \left( p(s^{l+1} = s^{l+1}(n_{(l,0)})) |s' = n_{(l,0)}| \right) E_{(1,1)} - 1 \]

(3)

where (see (4))

Therefore, in the case of $n_{(l,0)}$, we can know that if $p(s^{l+1} = s^{l+1}(n_{(l,0)})) |s' = n_{(l,0)}| < 1$, then the leaf-to-leaf transition method is more efficient than the root-to-root transition method. The condition can be modified like $p(s^{l+1} = s^{l+1}(n_{(l,0)})) |s' = n_{(l,0)}| < (1/E_{(1,1)})$, and we can expect that the leaf-to-leaf transition method is generally efficient than the root-to-root transition method considering that $E_{(1,1)}$ is generally large number when the hierarchical template matching is used with a template tree having a large number of branches and levels. In the section of experimental results, $C_{\text{leaf-to-leaf}}$ measured with a pedestrian silhouette database will be provided.

### 3 Internal-to-internal transition method

This paper proposes that when we are trying to detect a pedestrian in consecutive images by a hierarchical template matching, we should consider not only leaf node-to-leaf node transitions, but also internal node-to-internal node transitions to improve the efficiency of the tree traversal. Such a method is referred to as the internal-to-internal transition method, which could be regarded as a generalisation method including previously explained methods, the root-to-root transition method and the leaf-to-leaf transition method. The proposed internal-to-internal transition method has the same procedure as the leaf-to-leaf transition method as described by the pseudo code steps 1–3, except that the prediction $s^{l+1}(n_{(l,0)})$ can be selected from any node of a template tree including the internal nodes.

The internal-to-internal transition method is based on the fact that a parent–child relation of a template tree is not a decomposition relation, but an inheritance relation. For example, let us consider a situation where $n_{(5,a)}$ is matched in the time step $t$ and $n_{(5,b)}$ is matched in the time step $t + 1$ by a hierarchical template matching as shown in Fig. 3. If the leaf node $n_{(a)}$ is matched, it means that every nodes on the path from the root node to $n_{(5,a)}$, which is referred to as the direct link, are selected as the most appropriate node in each level. In other words, matched $n_{(5,a)}$ means every node of its direct link is, in a sense, matched. Such a characteristic coincides with a hierarchical finite state machine, where if a sub-state is active it means every state containing the sub-state is also active. Therefore, by regarding a child node as the sub-state of its parent node, a hierarchical template tree can be regarded as a hierarchical finite state machine and a state transition of the FOM model can be regarded as a state transition of a hierarchical finite state machine.

\[
E_{(l,0)} = \begin{cases} 
1, & \text{if } l = L \\
|C(n_{(l,0)})|, & \text{if } l = L - 1 \\
|C(n_{(l,0)})| + \sum_{n_{(l+1,0)} \in E(n_{(l,0)})} p(n_{(l+1,0)},n_{(l,0)}) E_{(l+1,0)}, & \text{otherwise}
\end{cases}
\]

\[
C_{\text{leaf-to-leaf}} = \sum_{n_{(l,0)} \in N_l} p(n_{(l,0)}) \left( p(s^{l+1} = s^{l+1}(n_{(l,0)})) |s' = n_{(l,0)}| + (1 - p(s^{l+1} = s^{l+1}(n_{(l,0)})) |s' = n_{(l,0)}|) (1 + E_{(1,1)}) \right)
\]

\[
C_{\text{root-to-root}} = E_{(1,1)} = \sum_{n_{(l,0)} \in N_l} p(n_{(l,0)}) E_{(1,1)} = \sum_{n_{(l,0)} \in N_l} p(n_{(l,0)}) = 1
\]

© The Institution of Engineering and Technology 2014

A hierarchical finite state machine can define every transition from a state of any level to a state of any level. Once we regard a hierarchical template tree as a hierarchical finite state machine, we should consider every transition from a node in any level to a node in any level as the candidate of state transition of the FOM model. All possible state transitions are depicted in Fig. 4. For this reason, although the leaf-to-leaf transition method assumes that the state variable \( \hat{s} \) is one of the leaf nodes, the internal-to-internal transition method extends the concept by regarding \( \hat{s} \) as one of all template tree nodes without any constraint on the level.

If an FOM model is adopted, the next time step node of a node is predicted using only the state of the current time step. Therefore the next state prediction of a node \( n_{(l,j)} \) of a hierarchical template tree including the internal nodes, \( \hat{s}^{t+1}(n_{(l,j)}) \), can be calculated as

\[
\hat{s}^{t+1}(n_{(l,j)}) = \arg \min_{n_{(m,j)} \in \mathbb{D}_1 \cup \mathbb{D}_2} \left\{ \min_{n_{(l,k)} \in \mathbb{D}_1} \left( p(\hat{s}^{t+1} = n_{(m,j)} \mid s' = n_{(l,k)}) \varepsilon_{(m,j)} \right) \right. \\
+ \left. \left( 1 - p(\hat{s}^{t+1} = n_{(m,j)} \mid s' = n_{(l,k)}) \right) \left( \varepsilon_{(m,j)} + \varepsilon_{(1,1)} \right) \right\}
\]  

(5)

where \( \mathbb{D}_1 \) denotes a node set consisting of nodes from level 1 to \( L \), that is the universal set of nodes of a hierarchical template tree. \( p(\hat{s}^{t+1} = n_{(m,j)} \mid s' = n_{(l,j)}) \) is the probability that the transition from the node \( n_{(l,j)} \) to \( n_{(m,j)} \) is correct, and the corresponding cost is \( \varepsilon_{(m,j)} \). \( \varepsilon_{(1,1)} \) is the number of template matching conducted from \( \hat{s}^{t+1}(n_{(l,j)}) \), that is \( n_{(m,j)} \), to the leaf level by a hierarchical template matching. In the same manner, the probability that the prediction is incorrect is \( 1 - p(\hat{s}^{t+1} = n_{(m,j)} \mid s' = n_{(l,j)}) \), and the corresponding cost is \( \varepsilon_{(m,j)} + \varepsilon_{(1,1)} \).

By modifying the next state prediction of a leaf node, \( \hat{s}^{t+1}(n_{(l,0)}) \), using the direct link concept, the next state prediction of the internal-to-internal transition method can be defined as

\[
\hat{s}^{t+1}(n_{(l,0)}) = \arg \min_{n_{(m,j)} \in \mathbb{D}_1} \left\{ \min_{n_{(l,k)} \in \mathbb{D}_1} \left( p(\hat{s}^{t+1} = n_{(m,j)} \mid s' = n_{(l,k)}) \varepsilon_{(m,j)} \right) \right. \\
+ \left. \left( 1 - p(\hat{s}^{t+1} = n_{(m,j)} \mid s' = n_{(l,k)}) \right) \left( \varepsilon_{(m,j)} + \varepsilon_{(1,1)} \right) \right\}
\]  

(6)

where \( \mathbb{D}_1 \) denotes the direct link of a leaf node \( n_{(l,0)} \), which are nodes on the path from the root node to the leaf node. For example, in Fig. 4, \( \mathbb{D}_1 \) is \( \{ n_{(1,a_1)}, n_{(2,a_2)}, n_{(3,a_3)}, n_{(4,a_4)}, n_{(5,a_5)} \} \). As in (6), by setting \( \hat{s}^{t+1}(n_{(l,0)}) \) to the one requiring the lowest cost among the next state predictions of the direct link determined by (5), we can reduce more cost than when considering only the leaf-to-leaf transitions. Note that the next state prediction of a leaf node \( n_{(l,0)} \), \( \hat{s}^{t+1}(n_{(l,0)}) \), is defined by (1) in the leaf-to-leaf transition method and modified by (6) in the internal-to-internal transition method.

Fig. 5 shows an example of the next state prediction considering the internal nodes. Let us assume that the leaf node \( a_3 \) is matched in the time step \( t \), and the next state prediction of \( a_3 \) is established in advance as ‘from \( a_3 \) to \( c_3 \).’ Once the next image is inputted, a hierarchical template matching is applied to a sub-tree of which the root is \( c_3 \), and let us assume that \( c_5 \) is detected as the most optimal node. If the matching error of \( c_5 \) is above a threshold, the hierarchical template matching is again applied to the root node. The main difference with the internal-to-internal
method is that the root-to-root transition method and leaf-to-leaf transition method firstly investigates $b_l$ and $b_s$, respectively. The cost of the internal-to-internal transition method, $C_{\text{internal-to-internal}}$ is defined as (see (7))

$$C_{\text{internal-to-internal}} = \sum_{n_{(l,k)} \in N_l} p(n_{(l,k)}) E_{(1,1)} - \sum_{n_{(l,k)} \in N_l} p(n_{(l,k)})$$

$$\times \left( p \left( s^{t+1} = n_{(m,j)} | s^t = n_{(l,k)} \right) E_{(m,j)} + \right)$$

$$\left( 1 - p \left( s^{t+1} = n_{(m,j)} | s^t = n_{(l,k)} \right) \right) \left( E_{(m,j)} + E_{(1,1)} \right)$$

$$= \sum_{n_{(l,k)} \in N_l} p(n_{(l,k)}) \sum_{n_{(m,j)} \in N_l} \left( E_{(1,1)} - p \left( s^{t+1} = n_{(m,j)} | s^t = n_{(l,k)} \right) E_{(m,j)} \right)$$

$$- \sum_{n_{(l,k)} \in N_l} p(n_{(l,k)}) \sum_{n_{(m,j)} \in N_l} \left( E_{(1,1)} - p \left( s^{t+1} = n_{(m,j)} | s^t = n_{(l,k)} \right) \right) \left( E_{(m,j)} + E_{(1,1)} \right)$$

$$= \sum_{n_{(l,k)} \in N_l} \left( E_{(1,1)} - p \left( s^{t+1} = n_{(m,j)} | s^t = n_{(l,k)} \right) \right) \left( E_{(m,j)} + \right)$$

$$\left( 1 - p \left( s^{t+1} = n_{(m,j)} | s^t = n_{(l,k)} \right) \right) \left( E_{(m,j)} + E_{(1,1)} \right)$$

$$= \sum_{n_{(l,k)} \in N_l} \left( E_{(1,1)} - p \left( s^{t+1} = n_{(m,j)} | s^t = n_{(l,k)} \right) \right) \left( E_{(m,j)} + \right)$$

$$\left( 1 - p \left( s^{t+1} = n_{(m,j)} | s^t = n_{(l,k)} \right) \right) \left( E_{(m,j)} + E_{(1,1)} \right)$$

(8)

It can be observed that, about a node $n_{(l,k)}$, if

$$p \left( s^{t+1} = n_{(m,j)} | s^t = n_{(l,k)} \right) E_{(1,1)} - E_{(m,j)} > 0,$$

then the internal-to-internal transition method is more efficient than the root-to-root transition method. In the condition, if we restrict $n_{(l,k)}$ and $n_{(m,j)}$ to a leaf node, then $E_{(m,j)} = 1$ and the condition becomes the same as that of the leaf-to-leaf transition method. From $C_{\text{internal-to-internal}} - C_{\text{leaf-to-leaf}} > 0$, we can find the condition when the internal-to-internal transition method is more efficient than the leaf-to-leaf transition method as below

$$p \left( s^{t+1} = n_{(m,j)} | s^t = n_{(l,k)} \right) E_{(m,j)} - E_{(1,1)} > 0.$$

When implementing the proposed method, various probabilities are used. Although they can be theoretically derived from the application-specific domain knowledge, they could be approximated by the values measured with the learning samples. Owing to the characteristics of a hierarchical template tree, by defining the number of learning sample belonging to each node, various probabilities can be measured. The learning sample subset belonging to the $l$th node of the $n_{(l,k)}$ level, $n_{(l,k)}$, is denoted by $\mathbb{S}(n_{(l,k)})$ and it is recursively defined as

$$\mathbb{S}(n_{(l,k)}) = \left\{ \begin{array}{l}
\{ n_{(l,k)} \}, \quad \text{if } l = L \n \bigcup_{n_{(l+1,k)} \subseteq \mathbb{S}(n_{(l,k)})} \mathbb{S}(n_{(l+1,k)}), \quad \text{otherwise}
\end{array} \right\}$$

(9)

where $\mathbb{S}(n_{(l,k)})$ denotes the set of the child node of $n_{(l,k)}$. The learning sample subset is used instead of the node set belonging to a sub-tree for two reasons. First, an internal node of a template tree is not an actual object, but only represents its sub-tree. Therefore it should be ignored when variation is considered. Second, as the level corresponding to the learning samples could be eliminated during the template tree construction as in [1], it cannot be assumed that the resultant leaf-nodes have the same occurrence frequency.

Using the defined $\mathbb{S}(n_{(l,k)})$, a priori probability of a node $n_{(l,k)}$, $p(n_{(l,k)})$, can be approximated as

$$\hat{p}(n_{(l,k)}) = \frac{|\mathbb{S}(n_{(l,k)})|}{|\mathbb{S}(n_{(1,1)})|}$$

(10)

where $|\mathbb{S}(n_{(1,1)})|$ is the same as the total number of learning samples. In the same manner, the probability, $p(n_{(l+1,k)} | n_{(l,k)})$, that the hierarchical template matching progresses to a specific child node after $n_{(l,k)}$, which is used to define ETMN $\mathbb{E}(l,k)$, can be approximated as

$$\hat{p}(n_{(l+1,k)} | n_{(l,k)}) = \frac{|\mathbb{S}(n_{(l+1,k)})|}{|\mathbb{S}(n_{(l,k)})|}$$

(11)

where the learning sample set belonging to each child node is assumed to be exclusive.

4 Experimental results

The proposed method is verified by applying to the detection of a pedestrian walking from side to side. Pedestrian detection has been one of the key problems of computer vision [8] and it also gains exponential interest from the field of advanced driver assistance system [9]. In particular, as the most dangerous situation is when a pedestrian pops out from behind a parked vehicle and crosses the road, an active pedestrian protection system focuses on pedestrians walking from side to side [10].

The experiments use 4617 pedestrian images of dataset A of the CASIA Gait database [11]. In each image, one pedestrian walks from side to side, and 20 persons are captured four times, respectively, and the length of each sequence is from 37 to 127 images. These sequences are
divided into two sets such that images and persons are exclusive to each other: the learning sample set consists of 3110 images (52 sequences) and the test sample set consists of 1507 images (28 sequences).

For each image, the foreground corresponding to the pedestrian is manually marked with an interactive segmentation tool [12], and the foreground is normalised to 80 × 80 in size such that the height of the pedestrians becomes the same. Finally, the pedestrian silhouette is extracted by applying a canny edge detector to the normalised foreground image. Fig. 6 shows an example of a pedestrian image, foreground image and pedestrian silhouette. The dissimilarity between two pedestrian silhouettes \( \mathcal{X} \) and \( \mathcal{Y} \), \( D(\mathcal{X}, \mathcal{Y}) \), is defined by a distance transform-based template matching score, frequently called chamfer distance [1] or modified Hausdorff distance [13], as

\[
D(\mathcal{X}, \mathcal{Y}) = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} \min\left(\frac{d_x}{dx}, \frac{d_y}{dy}\right) \left\| x + \left[\begin{array}{c} dx \\ dy \end{array}\right] - y \right\|_2
\]

where \( x \) and \( y \) are the edge pixel coordinates of \( \mathcal{X} \) and \( \mathcal{Y} \), respectively and \( dx \) and \( dy \) denote the \( x \)-axis and \( y \)-axis translations between two silhouettes. Rotational transformation is ignored during the matching as all pedestrians in the database are assumed to be upright.

A hierarchical template tree is constructed with the learning sample set by recursively applying a \( k \)-medoid clustering algorithm from the bottom level to the top level as proposed by Gavrila [1] (see [14]). The structure of the hierarchical template tree is similar to [1]: it consists of five levels and each level has 1, 8, 64, 512 and 3110 nodes, respectively and the bottom level is removed from the tree for the generalisation as suggested by Gavrila [1]. When \( k \)-medoid clustering algorithm is applied to each level, the maximum iteration number is set to a sufficiently large number \( (>100) \) and the simulated annealing is applied to avoid the local minima. Hierarchical template matching using the constructed hierarchical template tree follows the method selecting the best branch at each level from the root node to the leaf level [15].

To each leaf node of the hierarchical template tree, information about which node is predicted to be matched in the next time step after the node is attached according to the leaf-to-leaf transition method and the internal-to-internal transition method. As the hierarchical template tree is constructed with the learning sample set, one node of the hierarchical template tree corresponding to each image of the learning sample set can be determined. Additionally, as each sequence of the learning sample set has information about which pedestrian silhouette will appear after a specific pedestrian silhouette, the transition probability between leaf nodes can be measured. In the same manner, as the transition probability between the direct links of each leaf node can be measured with consecutive leaf node pairs, the transition probability between the internal nodes can be measured too. As the learning sample subset corresponding to a node of the hierarchical template tree can be determined by (9) and the branching probability to each child node of a node can be measured by (11), the ETMN of each node can be measured. Consequently, by (1) and (6), the next time step node of each node is determined according to the leaf-to-leaf transition method and the internal-to-internal transition method.

Table 1 compares the results of the learning and test phase of three transition methods. Owing to space considerations, root-to-root, leaf-to-leaf and internal-to-internal is abbreviated to R2R, L2L and I2I, respectively. Columns of each method shows the dissimilarity threshold determining whether a transition prediction is acceptable or not, the probability that a transition prediction succeeds, transition cost and average dissimilarity, in consecutive order. As R2R always starts the matching from the root node, the probability that a transition prediction succeeds is not defined and as the probability and the average dissimilarity are evaluated with the whole of the test sample set, they are not defined for the learning phase. Note that the transition cost of the learning and test phases has a somewhat different meaning: the transition cost of the learning phase is estimated with the learning sample set and that of the test phase is measured with the test sample set.

During the learning phase, the dissimilarity threshold determining whether a prediction is acceptable or not is set to 0.800. The threshold is heuristically determined and it is a dissimilarity when two silhouettes seem almost to overlap by the human eye. About 34.63% of leaf nodes set their

<table>
<thead>
<tr>
<th>Method</th>
<th>Threshold</th>
<th>Probability, %</th>
<th>Cost</th>
<th>( D )</th>
</tr>
</thead>
<tbody>
<tr>
<td>R2R</td>
<td>learning</td>
<td>0.800</td>
<td>-</td>
<td>26.90</td>
</tr>
<tr>
<td></td>
<td>test1</td>
<td>0.800</td>
<td>-</td>
<td>26.14</td>
</tr>
<tr>
<td></td>
<td>test2</td>
<td>0.800</td>
<td>-</td>
<td>26.14</td>
</tr>
<tr>
<td>L2L</td>
<td>learning</td>
<td>0.800</td>
<td>-</td>
<td>16.59</td>
</tr>
<tr>
<td></td>
<td>test1</td>
<td>0.800</td>
<td>-</td>
<td>19.14</td>
</tr>
<tr>
<td></td>
<td>test2</td>
<td>0.875</td>
<td>-</td>
<td>16.11</td>
</tr>
<tr>
<td>I2I</td>
<td>learning</td>
<td>0.800</td>
<td>-</td>
<td>15.41</td>
</tr>
<tr>
<td></td>
<td>test1</td>
<td>0.800</td>
<td>-</td>
<td>20.44</td>
</tr>
<tr>
<td></td>
<td>test2</td>
<td>0.985</td>
<td>-</td>
<td>12.77</td>
</tr>
</tbody>
</table>
next time step node prediction to an internal node instead of a leaf node as their internal-to-internal transition cost is smaller than their leaf-to-leaf transition cost. As we expected, with the whole of the learning sample set, the leaf-to-leaf transition cost is significantly smaller than the root-to-root transition cost, and the internal-to-internal transition cost is smaller than the leaf-to-leaf transition cost.

Fig. 7 shows an example when the prediction is set to an internal node. After the leaf node <4, 165> is matched in the time step \( t \), the leaf nodes <4, 75>, <4, 148> and <4, 165> are matched in the next time step \( t + 1 \) with the probability of 53.3, 13.3 and 33.3%, respectively. The leaf node <4, 75> that is when the pedestrian’s forward stepping progresses for a moment has the highest probability and the leaf node <4, 165> that is when the pedestrian maintains almost the same pose also appears. As these three leaf nodes are the child nodes of the node <3, 49>, the probability that the node <3, 49> will be matched after the node <4, 165> is estimated at 100%. As the node <3, 49> has 11 child nodes, the transition cost to the node <3, 49> is calculated as 11 and as it is the smallest among the transition costs of the direct link of the leaf node <4, 165>, it is determined as the internal-to-internal transition cost \( C_{\text{internal-to-internal}} \).

Meanwhile, the transition cost to the node <4, 75>, of which leaf-to-leaf transition probability is the highest, is the leaf-to-leaf transition cost \( C_{\text{leaf-to-leaf}} \) and calculated as 13.55. Consequently, we can expect that it will be the most efficient that the matching in the next time step starts from the node <3, 49> after the leaf node <4, 165> is matched. Therefore, the transition prediction of the leaf node <4, 165> is set to the node <3, 49>.

Generally, when a pedestrian is detected in consecutive images, it is well known that the position and scale can be successfully tracked by the Kalman filtering [9]. As the same hierarchical template matching will be iteratively applied to the neighbourhood of the estimated position and scale, we focus only on the searching within the hierarchical template tree during the test phase. Once an image of the test sample set is given, assuming a pedestrian is successfully detected in the image as in [3], we compare the amount of template matching to detect the pedestrian in the image of the next time step. In other words, the test sample image is matched with all leaf nodes of the hierarchical template tree and is assumed to be matched with the node having the smallest dissimilarity. Then, the number of matching is measured until a leaf node of the hierarchical template tree corresponding to the pedestrian silhouette of the next time step image is found. Such a test method is devised to exclude the effects of the performances of the constructed hierarchical template tree and tracking method, and to focus on the reduction of the template matching number with consecutive images.

Test 1 represents the test phase which uses the same dissimilarity threshold (=0.800) as the learning phase. Although the leaf-to-leaf transition cost is reduced by 26.7% compared with the root-to-root transition cost, contrary to our expectations, the internal-to-internal transition cost is rather larger than the leaf-to-leaf transition cost. However, instead, we need to note that the internal-to-internal transition method shows significantly smaller average dissimilarity than the other two methods: when the average dissimilarity of the leaf-to-leaf transition method is smaller than that of the root-to-root transition method by 0.006, the average dissimilarity of the internal-to-internal transition method is smaller than that of the leaf-to-leaf transition method by 0.011. From the observation, we can infer that the internal-to-internal transition method can find more proper nodes than the leaf-to-leaf transition method.

Test 2 represents the test phase which, to fairly compare the transition costs, adjusts the dissimilarity thresholds such that the average dissimilarity of each method has the same value. In other words, to make the average dissimilarity of the leaf-to-leaf transition method and the internal-to-internal transition method equal to that of the root-to-root transition method (in this case, 0.833), the dissimilarity threshold of
each method are changed to 0.875 and 0.985, respectively. Consequently, compared with the root-to-root transition cost, the leaf-to-leaf transition cost is reduced by 38.37% and the internal-to-internal transition cost is reduced by 51.15%. These two reduction ratios are the previously mentioned $\gamma_{\text{leaf-to-leaf}}$ and $\gamma_{\text{internal-to-internal}}$ in the form of ratio to the root-to-leaf transition cost. In particular, we need to note that the internal-to-internal transition cost is reduced by 20.73% compared with the leaf-to-leaf transition cost.

By aggregating the results of test 1 and test 2, we can infer that the proposed FOM-based prediction of the next time step node can either enhance the performance of the hierarchical template matching or significantly reduce the number of matching operations. In particular, it is confirmed that the internal-to-internal transition method, newly proposed in this paper, shows better performance than the leaf-to-leaf transition method. It is coincident with the analysis of [13] that the middle level node of a hierarchical template tree is more reliable than the root node or the leaf level node.

The previous methods starting from the root node in each time step can be regarded as a root-to-root transition method no matter whether they search multi-path or not. As it is certain that the cost of a single-path searching is smaller to that of a multi-path searching, the comparison between the transition methods using only the single-path searching is thought to be reasonable to show the contribution of the proposed method. If we need to adopt the multi-path searching, we can firstly investigate the most probable node in the single-path searching manner and, if the search fails, we can go to the root node in the multi-path searching manner. Considering that multi-path searching will need a significantly larger cost than single-path searching, the proposed method is expected to be effective in the multi-path searching. However, exploiting the information of all matched nodes during multi-path searching as well as the finally determined leaf node seems to require further study.

The proposed method assumes that only one pedestrian exists and the position and scale are successfully tracked. Therefore further study is needed to deal with when region of interest (ROI) tracking fails or when multiple hypotheses are simultaneously considered as in particle filtering. Additionally, as the transition probability between nodes are supposed to change according to the pedestrian speed, a method alternating the prediction considering the speed is supposed to be possible. As a template matching can be regarded as a weak classifier as mentioned in [13], the proposed method could be applied to a hierarchical classifier dealing with temporally related data.

5 Conclusions

This paper proposes, when a hierarchical template matching is consecutively applied to the detection of an object continuously changing its appearance or the estimation of its pose, a method firstly investigating the most probable node in the next time step. By applying the proposed method to the pedestrian silhouette detection, it is confirmed that the proposed internal-to-internal transition method can reduce the matching error or cost compared with the root-to-root transition method, which always starts from the root node and the leaf-to-leaf transition method, which considers only the transition between leaf nodes. In particular, when making the matching error of each method equal, the cost of the leaf-to-leaf transition method and internal-to-internal transition method is reduced by 38.37 and 51.15%, respectively, compared with the cost of the root-to-root transition method. As a template matching is a kind of weak classifier, the proposed method could be applied to a hierarchical classifier structure. Further studies are required to deal with multi-path searching, ROI tracking failure and adaptation to pedestrian’s speed.

6 Acknowledgment

This research was supported by the Basic Science Research Programme through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (no. 2012-0002467).

7 References

11 Gait Database. Available at http://www.cbsr.iia.ac.cn/english/Gait%20Databases.asp