Human Gait Identification using a Particle Filter

Mitsuharu Emoto, Akira Hayashi, Nobuo Suematsu and Kazunori Iwata
Faculty of Information Sciences, Hiroshima City University
Ozuka-Higashi 3-4-1, Asaminami-Ku, Hiroshima, 731-3194, JAPAN
Email: {emoto, akira, suematsu, kiwata}@robotics.im.hiroshima-cu.ac.jp

Abstract—We approach the human identification problem from the perspective of gait and body shape. Conventional methods depend on the camera viewing direction, and since they are based on matching image silhouettes or features their identification accuracy is low when there is a big difference between the camera viewing direction of the test and training data. Thus, if a person is walking in an arbitrary direction, they may not be accurately identified.

In this paper, we propose a novel method that does not depend on the camera viewing direction. We develop a state space model called a “cyclic motion model” whose state variables are not only the phase of the motions but also the camera viewing direction. We learn model parameters for each candidate person, and represent their walking with the cyclic motion model. To identify a person from the observed image sequence, we first compute the model likelihoods for the sequence using a particle filter that represents a probability distribution by a set of weighted samples. We then identify the person from model likelihoods.

I. INTRODUCTION

Human identification from biological information such as fingerprint, retina, etc. has often been used recently for surveillance. Such identification methods are accurate, but need special scanner systems and people must make contact with the system. Though people appear to walk similarly, individual people’s walking styles show characteristics of gait and body shape. Thus walking image sequences may have sufficient information to identify a person [1], who to be identified does not have to be scanned voluntarily, since we can obtain walking data easily utilizing a commercial video camera.

In conventional methods, a person is identified from an image sequence which is taken by a fixed monocular camera when that person is walking in a known direction [1], [2], [4], [5]. If a person is walking in an arbitrary direction, they may not be accurately identified. Since these methods are based on matching image silhouettes or features, identification accuracy is low when there is a big difference between the camera views of the test and training data. Therefore, these methods depend on the camera viewing direction. However, in the real world it is important to identify a person without depending on a particular camera viewing direction.

Using a Bilinear model [12], Cuzzolin et al. estimated an unknown camera viewing direction from which a image sequence is taken when a person is walking and then they identified the person [6]. When the walking person makes a turn, the viewing direction changes, but they cannot estimate the changed viewing direction. Shakhnarovich et al. also can identify a person from a image sequence that is taken from various viewing directions that change with time, but they needed multiple video cameras [7].

In this paper, we propose a novel method that does not depend on the camera viewing direction. We develop a state space model called the “cyclic motion model” (CMM), in which the state variables are the phases of the motion and the camera viewing direction, and the observed variables are image features taken by a fixed camera. We learn the model parameters for each candidate person, and represent their walking using the cyclic motion model. To correctly identify a person from the observed image sequence, we first compute the model likelihoods for the sequence using a particle filter that represents a probability distribution from a set of weighted samples. We then identify the person from the model likelihoods.

We present the problem setting for the identification in Sec. II. We show the schema of the proposed method in Sec. III, and the details are explained in Sec. IV and Sec. V. In the experimental Sec. VI, we report and discuss the efficiency and the accuracy of our method. Finally, Sec. VII deals with conclusions.

II. PROBLEM SETTING

We identify a k-th person who is one of K known people from an image sequence that is taken when that person is walking in an arbitrary direction. We denote the image sequence by \( y_{1:t} = (y_1, ..., y_t) \) where \( y_i \) is an image feature vector at time \( t \).

We assume that we can obtain in advance the image sequences of the \( K \) people that are taken from \( V \) fixed camera viewing directions: \( \{D_{(k, v_i)} \mid 1 \leq k \leq K, 1 \leq i \leq V \} \) as training data, where

\[
D_{(k, v_i)} = (\tilde{y}_{1|v_i}, \tilde{y}_{2|v_i}, ..., \tilde{y}_{T|v_i})
\]

\( \{v_i \mid 0 \leq v_i < 2\pi, 1 \leq i \leq V \} \) is a camera viewing direction and \( \tilde{y}_{t|v_i} \) is an image feature vector which is obtained from \( v_i \) at time \( t \).

III. PROPOSED METHOD

A. State space model

In the CMM, we assume a cyclic motion, thus the state variable of the CMM is \( \varphi_t \) where \( \varphi_t \in (0, 2\pi) \) is the phase of a cyclic motion [13]. Adding \( \psi_t \in (0, 2\pi) \) which denotes the camera viewing direction to the state variable vector \( \varphi_t \), we can deal with the changing of the camera viewing direction (e.g. from front to right) within the sequence.
We denote the state variable and the observed variable (i.e., the image feature vector) at time \( t \) as \( x_t = (\phi_t, \psi_t)^T \) and \( y_t \) respectively. The state equations in the CMM is as follows:

\[
\begin{align*}
    x_t &\sim \Pr(x_{t-1}, \lambda_k) \\
y_t &\sim \Pr(y|x_t, \lambda_k)
\end{align*}
\]

where \( \lambda_k \) is the model for the \( k \)-th person.

**B. Algorithm for the proposed method**

The proposed method consists of two steps.

- **Learning step**
  1) define the phase of each cyclic motion from the training data.
  2) learn the model parameters for each model \( \lambda_k \): the state transition probability \( \Pr(x|x_{t-1}, \lambda_k) \) and the observation probability \( \Pr(y|x_t, \lambda_k) \).
  3) Identify the person from \( \{\Pr(\lambda_k)|1 \leq k \leq K\} \).

**IV. LEARNING STEP**

**A. Defining the phase**

We have to define the phase of each motion from the training data \( D_{(k,v)} \) since we exploit characteristics of cyclic motion. Since the velocity at which a person walks is not always constant, we define the phase using the following procedure. Here, a subscript \( i \) of \( v_i \) is omitted for brevity.

1) The training data \( D_{(k,v)} \) is divided into \( M \) subsequences such that each sub-sequence corresponds to a cycle. If a sub-sequence that is not enough for one cycle exists, we delete it.

\[
D_{(k,v)} = \left( \tilde{y}_{1|v}, \ldots, \tilde{y}_{T_1|v}, \tilde{y}_{T_1+1|v}, \ldots, \tilde{y}_{T_1+T_2|v}, \ldots, \tilde{y}_{\sum_{m=1}^{M-1} T_m+1|v}, \ldots, \tilde{y}_{\sum_{m=1}^{M-1} T_m+T_m+1|v} \right)
\]

\[
= \left( D_{(1),k,v}, D_{(2),k,v}, \ldots, D_{(M),k,v} \right)
\]

where \( T_i \) is set to \( T \geq \sum_{i=1}^{M} T_i \).

2) Using Dynamic Time Warping (DTW) \cite{8}, we obtain the correspondence of frames between \( D_{(i),k,v} \) and \( D_{(j),k,v} \) and the DTW distance \( \{d^2(D_{(i),k,v}, D_{(j),k,v}) | 1 \leq i, j \leq M, i \neq j\} \).

3) We select a prototype cycle \( D_{(P),k,v} \) from \( M \) cycles.

\[
P_k = \arg \min_{1 \leq i \leq M} \sum_{j=1}^{M} d^2(D_{(i),k,v}, D_{(j),k,v}).
\]

4) The phase \( \{\phi_t | 1 \leq t \leq T_{P_k}\} \) for each frame in the prototype cycle \( D_{(P),k,v} \) is defined as follows.

\[
\phi_t = \frac{2\pi}{T_{P_k}}(t - 1)
\]

5) The phase for each frame in non-prototype cycles \( \{D_{(i),k,v} | 1 \leq i \leq M, i \neq P_k\} \) is determined to be the phase of the corresponding frame in the prototype cycle.

**B. Learning the model parameters**

The state transition probability \( \Pr(x|x_{t-1}, \lambda_k) \) is approximated by the Gaussian \( N(\mu_{x}^{(k)}, \Sigma_{x}^{(k)}) \). We assume that the phase \( \phi_t \) proceeds as the prototype \( D_{(P),k,v} \) and that the camera viewing direction \( \psi_t \) changes slowly. We also assume there is no correlation between the phase and the camera viewing direction.

\[
\begin{align*}
    \mu_{x}^{(k)} &= (\phi_{t-1} + 2\pi/T_{P_k}, \psi_{t-1})^T \\
    \Sigma_{x}^{(k)} &= \text{diag}\left((2\pi/T_{P_k})^2, (\epsilon_v^{(k)})^2\right)
\end{align*}
\]

where \( \epsilon_v^{(k)} \) is the variance in the camera viewing direction that is determined experimentally.

The observation probability \( \Pr(y|x_t, \lambda_k) \) is approximated by the Gaussian \( N(\mu_{y}^{(k)}, \Sigma_{y}^{(k)}) \). We learn (and interpolate) \( \mu_{y}^{(k)} \) from training data \( \{D_{(k,v)} | 1 \leq k \leq K, 1 \leq i \leq V\} \) using the radial basis function (RBF) network \( f_k(x_t) \):

\[
\mu_{y}^{(k)} = f_k(x_t)
\]

The covariance matrix \( \Sigma_{y}^{(k)} \) is determined empirically.

Now, we have all the parameters for the model \( \lambda_k \):

\[
\lambda_k = (\mu_{x}^{(k)}, \Sigma_{x}^{(k)}, \mu_{y}^{(k)}, \Sigma_{y}^{(k)})
\]

**V. IDENTIFICATION STEP**

To identify a person from the observed image sequence \( y_{1:T} \), we first compute the model likelihoods \( \{\Pr(\lambda_k)|1 \leq k \leq K\} \) for the sequence. We then estimate the identification \( k^* \) from the model likelihoods.

\[
k^* = \arg \max_{1 \leq k \leq K} \Pr(\lambda_k)
\]

The model likelihood \( \Pr(\lambda_k) \) is defined as follows (we omit subscript \( k \) of \( \lambda_k \) for brevity).

\[
\Pr(\lambda_k) = \prod_{t=1}^{T} \Pr(y_t | y_{1:t-1}, \lambda_k)
\]

where

\[
\Pr(y_t | y_{1:t-1}, \lambda_k) = \int \Pr(y_t | x_t, \lambda_k) \Pr(x_t | y_{1:t-1}, \lambda_k)\, dx_t = \int \Pr(x_t | y_{1:t-1}, \lambda_k) \Pr(x_t | x_{t-1}, \lambda_k)\, dx_t dx_{t-1}
\]

\[
\Pr(x_t | x_{t-1}, \lambda_k) = \int \Pr(x_t | x_{t-1}, \lambda_k) \Pr(x_{t-1} | y_{1:t-1}, \lambda_k)\, dx_{t-1}
\]

\[
\Pr(x_{t-1} | y_{1:t-1}, \lambda_k) = \int \Pr(x_{t-1} | y_{1:t-1}, \lambda_k) \Pr(y_t | x_{t-1}, \lambda_k)\, dx_{t-1}
\]

\[
\Pr(y_{1:t-1} | \lambda_k) = \int \Pr(y_{1:t-1} | \lambda_k) \Pr(\lambda_k)\, d\lambda_k
\]

\[
\Pr(\lambda_k) = \prod_{t=1}^{T} \Pr(y_t | y_{1:t-1}, \lambda_k)
\]

where

\[
\Pr(y_t | y_{1:t-1}, \lambda_k) = \int \Pr(y_t | x_t, \lambda_k) \Pr(x_t | y_{1:t-1}, \lambda_k)\, dx_t = \int \Pr(x_t | y_{1:t-1}, \lambda_k) \Pr(x_t | x_{t-1}, \lambda_k)\, dx_t dx_{t-1}
\]

\[
\Pr(x_{t-1} | y_{1:t-1}, \lambda_k) = \int \Pr(x_{t-1} | y_{1:t-1}, \lambda_k) \Pr(y_t | x_{t-1}, \lambda_k)\, dx_{t-1}
\]

\[
\Pr(y_{1:t-1} | \lambda_k) = \int \Pr(y_{1:t-1} | \lambda_k) \Pr(\lambda_k)\, d\lambda_k
\]
A. Computing likelihoods using a particle filter

The particle filter is an approximate computation method for efficiently solving the state estimation problem [9]. It represents the posterior probability distribution by particles (weighted samples) that are distributed in the state space. We denote the \(i\)-th sample at time \(t\) as \(x_t^{(i)} = (\phi_t^{(i)}, \psi_t^{(i)})^T\), and the weight for the sample as \(w_t^{(i)}\). The posterior probability distribution \(Pr(x_t|y_{1:t}, \lambda)\) is computed approximately using a set of \(N\) weighted samples: \(X_{t|\lambda} = \{x_t^{(i)}, w_t^{(i)}|1 \leq i \leq N\}\). Here, \(w_t^{(i)}\) is set to satisfy \(\sum_{i=1}^N w_t^{(i)} = 1\).

\[
Pr(x_t|y_{1:t}, \lambda) \approx \sum_{i=1}^N w_t^{(i)} \delta_{x_t^{(i)}}(x_t) \tag{13}
\]

where

\[
\delta_{x_t^{(i)}}(x_t) = \begin{cases} 
1 & \text{if } x_t = x_t^{(i)} \\
0 & \text{otherwise}
\end{cases} \tag{14}
\]

Using eq. 13, \(Pr(y_t|y_{1:t-1}, \lambda)\) in eq. 12 is approximated as follows:

\[
Pr(y_t|y_{1:t-1}, \lambda) \approx \sum_{i=1}^N w_t^{(i)} \Pr(y_t|x_t^{(i)}, \lambda) \tag{15}
\]

where \(\{x_t^{(i)}|1 \leq i \leq N\}\) is a set of samples obtained from \(\{x_t^{(i)}|1 \leq i \leq N\}\) according to the state transition probability \(Pr(x_t|x_t^{(i)}_{t-1}, \lambda)\).

Thus the model likelihood \(L(\lambda)\) is approximated by the particle filter as follows:

\[
L(\lambda) = \sum_{t=1}^T \log Pr(y_t|y_{1:t-1}, \lambda) \\ \approx \sum_{t=1}^T \log \left(\sum_{i=1}^N w_t^{(i)} \Pr(y_t|x_t^{(i)}, \lambda)\right). \tag{16}
\]

B. Algorithm of the particle filter

Given \(X_{t-1|\lambda}\), we compute \(X_{t|\lambda}\) by repeating the following procedure \(N\) times.

1) resample a sample \(x_{t-1}^{(i)}\) at time \(t-1\) according to the distribution approximated \(X_{t-1|\lambda}\).

\[
x_{t-1}^{(i)} \sim X_{t-1|\lambda} \tag{17}
\]

2) generate a sample \(x_t^{(i)}\) at time \(t\) according to the state transition probability \(Pr(x|t-1, \lambda)\).

\[
x_t^{(i)} \sim Pr(x|x_t^{(i)}_{t-1}, \lambda) \tag{18}
\]

3) compute a weight \(w_t^{(i)}\) of the sample \(x_t^{(i)}\) using the observation probability \(Pr(y|x_t^{(i)}, \lambda)\).

\[
w_t^{(i)} = Pr(y|x_t^{(i)}, \lambda) \tag{19}
\]

The initiate sample set \(X_{0|\lambda}\) is sampled from the uniform distribution and all the weights are set to \(1/N\).

VI. EXPERIMENTS

To show the effectiveness of our method, we report experiments in which we compare our method with the Baseline algorithm that is one of the conventional methods.

The Baseline algorithm simply attempts to identify a person using matching scores from silhouette images [1]. The original Baseline algorithm cannot deal with the camera viewing directions that are different from that in the training data. So, in order to compare it fairly with our proposed method, we improve it so that its training data can contain image sequences taken from multiple viewing directions, and so the algorithm can select the best matching direction. The Baseline algorithm is explained in the appendix.

A. Data

We use ten people created by a commercial CG software [11] and shown in Fig. 1. We provide six kinds of gaits for them, and assigned these as follows: the first gait is to the 1st and 7th person, the second to the 3rd, 5th and 10th person, the third to the 6th and 8th person, the fourth to the 2nd person, the fifth to the 4th person, and the sixth to the 9th person.

1) Image feature vector: We use Hu’s seven moments [10] computed from silhouette images as \(y\), and there are invariant for scaling, translation and rotation.

2) Training data: We obtain the training data \(\{D(k,v_i) \mid 1 \leq k \leq 10, 1 \leq i \leq 8\}\) as follows. We let people walk on a treadmill, then take image sequences from the cameras. Four monocular cameras are placed around the treadmill (see fig. 2(a)): front (\(\psi = 0\)), diagonally forward left (\(\psi = \pi/4\)), left (\(\psi = \pi/2\)) and diagonally forward right (\(\psi = 7\pi/4\)). For silhouette images, we can synthesize a new image sequence that is taken from an opposite direction using a mirror reversed image sequence. Thus, we obtain image sequences from 8 directions, \(v_i \in \{(i-1)\pi/4 \mid 1 \leq i \leq 8\}\), and use these as the training data. We show parts of the training data for the 1st person in fig. 2(b). The training data is used to learn the model parameters \(\{\mu^{(k)}_x, \Sigma^{(k)}_x, \mu^{(k)}_y, \Sigma^{(k)}_y \mid 1 \leq k \leq 10\}\). \(\Sigma^{(k)}_x\) in \(\Sigma^{(k)}_x\) is set to \(2\pi/32\), and \(\Sigma^{(k)}_y = \text{diag}(0.3, 0.5, 1.8, 0.5, 1.3, 0.8, 0.4)\).

3) Test data: As test data, we use 50 image sequences taken when each of the ten people is walking along five kinds of walking paths as shown in Fig. 3(a). Parts of the test data are shown in fig. 3(b). The camera viewing direction with respect
to the walking person does not change for paths (1) and (2), but it changes for the other paths. The viewing direction for path (1) is contained in the training data, but that for path (2) is not. For brevity, we denote the test using the walking sequence along path (i) as test (i).

B. Results

In order to compute each model likelihood $L(\lambda_k)$, we provide 100 samples in the state space model for the model $\lambda_k$. That is, we use a total 1000 samples in the experiment for identifying one person from ten people. Table I shows the identification accuracy for both the proposed method and the Baseline algorithm. We also show parts of the model likelihoods in Fig. 4 when the 1st, 5th, and 7th persons walk along paths (1) and (3).

1) Tests (1) and (2): While the identification rate of the Baseline algorithm for test (2) is 70%, the identification rate of our method is 100% for both tests (1) and (2). While the Baseline algorithm makes mistakes because the camera viewing direction for path (2) is not contained in the training data, our method accurately learns and interpolates the novel camera view from the training data. This point is explained further in Sec. VI-C.

2) Tests (3), (4) and (5): In these tests, the camera viewing direction changes with time. The Baseline algorithm, which cannot deal with the change, shows low accuracy in comparison with our method (see Table I). In order to correctly identify people, we must accurately estimate the camera viewing direction, which is one of the state variables. Given the identification $k^\ast$, we estimate the state variables $\hat{x}_t = (\phi_t, \psi_t)^T$ from the mode of the posterior probability distribution $Pr(x_t|y_{1:t}, \lambda(k^\ast))$, that is, the sample set $X_t|\lambda(k^\ast)$.

$$\hat{x}_t \approx \text{mode} \left(X_t|\lambda(k^\ast)\right)$$ (20)

Fig. 5 shows the true and estimated values of the state variables for the 1st person in tests (4) and (5). We can see that the state variables are accurately estimated. Estimation errors of both variables are limited to the early time frames with insufficient observations, as errors decrease with time.
These sequences are used to obtain true values for $C$. Discussion

It mistakes the $7$th person for the $1$st person, and since they both have the same gait, and their body shapes are proportional. The Hu’s seven moments cannot distinguish these body shapes well, since the moments are scale invariant. The problem could be overcome by adding image features.

We have evaluated the accuracy of the interpolation for the mean parameter of the observation probability, $\mu_{M}^{(k)}(x_{t})$, using the RBF network $f_{k}(x_{t})$ (see eq.8). The number of hidden units in the network was 40. Using CG, we generate image sequences from various viewing directions: $\psi \in \{(i-1)\pi/4 \mid 1 \leq i \leq 8\} \cup \{(j-1)\pi/6 \mid 1 \leq j \leq 12\}$. These sequences are used to obtain true values for $f_{k}(x_{t})$.

We compute the relative error $Err$ as follows:

$$Err = \frac{1}{H} \sum_{h=1}^{H} \frac{||\text{u}_{h}^{\text{true}} - \text{u}_{h}^{\text{learned}}||}{||\text{u}_{h}^{\text{true}}||}.$$  

(21)

Where $\text{u}_{h}^{\text{true}}$ and $\text{u}_{h}^{\text{learned}}$ are the $\{|h| \mid 1 \leq h \leq H\}$-th true and learned RBF values, respectively. Table II shows that the relative error is small for all models.

\begin{table}[h]
\centering
\caption{Relative error from the RBF network}
\begin{tabular}{|c|c|c|c|c|c|}
\hline
person & 1 & 2 & 3 & 4 & 5 \\
\hline
Err & 0.114 & 0.103 & 0.130 & 0.100 & 0.097 \\
\hline
person & 6 & 7 & 8 & 9 & 10 \\
\hline
Err & 0.121 & 0.085 & 0.078 & 0.126 & 0.101 \\
\hline
\end{tabular}
\end{table}

D. Computation Time

The particle filter is often discussed with respect to the computational demand caused by the high dimensionality of the state space. However, we efficiently use the particle filter since we use CMM which is a low dimensional state space model and its state variable vector is two dimensional. Given a test sequence with a length of $M$ frames, we compute the model likelihoods $\{L(\lambda_{k})\mid 1 \leq k \leq K\}$ sequentially. The computation time $T_{p}$ is as follows.

$$T_{p} \propto nMK$$

(22)

where $n$ is the number of samples per one model. When we set $M = 120$, $n = 100$ and use MATLAB running on PowerMac G5 (Dual PowerPC 2.7GHz), the actual time is about 1 minutes.

In the Baseline algorithm, a person is identified by computing the matching score between the test sequence and each of the training sequences whose length is $N$ frames. When $N \approx M$, the computation time $T_{b}$ is as follows.

$$T_{b} \propto M^{2}KV$$

(23)

where $V$ is the number of camera viewing directions for the training data. The actual computation time is about 280 minutes. The Baseline algorithm, though the codes are not optimized well, creates a heavy computational load since it must scan all the image pairs of both the test and training data to compute a matching score.

VII. CONCLUSION

We have approached the problem of identifying a person who is one of a group of known people when that person is walking in an arbitrary direction. Many conventional methods depend on the camera viewing direction. And since they are based on matching image sequences, their identification accuracy is low when there is a big difference between the camera viewing directions of the test and training data.

We have proposed a method for identifying a person using a particle filter from gait and body shape, and have shown its effectiveness. We developed a state space model called the...
“cyclic motion model” that has state variables that are not only the phase of the motions but also the camera viewing directions. We learned the model parameters for each candidate person, and represented that persons walking with the cyclic motion model. To identify the person from the observed image sequence, we first computed the model likelihoods for the sequence using the particle filter, and then we identified the person from the model likelihoods.

REFERENCES


APPENDIX

BASELINE ALGORITHM

The Baseline algorithm simply attempts to identify a person using matching scores from silhouette images [1]. The method computes the matching score \( \text{Sim}(S_P, S_G) \) between the probe (test) silhouette image sequence \( S_P = \{ S_P(1), S_P(2), \ldots, S_P(M) \} \) and the gallery (training) silhouette image sequence \( S_G = \{ S_G(1), S_G(2), \ldots, S_G(N) \} \) as follows. Let us denote one cycle length of the probe \( S_P \) as \( N_{gait} \) and the probe sub-sequence as \( S_P = \{ S_P(t), S_P(t+1), \ldots, S_P(t+N_{gait}-1) \} \) \( 1 \leq t \leq M-N_{gait}+1 \).

1) Set \( \text{FrameSim}(S_P(i), S_G(j)) \) which is the score between the \( i \)-th probe image \( S_P(i) \) and the \( j \)-th gallery image \( S_G(j) \).

\[
\text{FrameSim}(S_P(i), S_G(j)) = \frac{\text{Num}(S_P(i) \cap S_G(j))}{\text{Num}(S_P(i) \cup S_G(j))}
\]

where \( \text{Num}(S) \) is the number of white pixels (which denote a person) in a silhouette image \( S \).

2) Compute \( \{ \text{Corr}(S_P(i), S_G(l)) | 1 \leq l \leq N-N_{gait}+1 \} \) which is the score between the probe subsequence \( S_P \) and the gallery sequence \( S_G \).

\[
\text{Corr}(S_P(i), S_G(l)) = \sum_{j=0}^{N_{gait}-1} \text{FrameSim}(S_P(t+j), S_G(l+j))
\]

3) Compute \( \text{Sim}(S_P, S_G) \).

\[
\text{Sim}(S_P, S_G) = \frac{1}{V} \sum_{1 \leq k \leq V} \max \text{Corr}(S_P, S_G(l))
\]

The above Baseline algorithm cannot deal with camera viewing directions that are different from that in the training data. In order to compare it fairly with the proposed method, we improve the baseline algorithm so that its training data can contain image sequences taken from multiple viewing directions, and so that the algorithm can select the best matching direction.

Let the training data for the \( k \)-th person from \( V \) directions be \( S_G = \{ D_{(k,v)} | 1 \leq k \leq K, 1 \leq i \leq V \} \). We define the matching score \( \text{Sim}_k(S_P, S_G) \) for the \( k \)-th person.

\[
\text{Sim}_k(S_P, S_G) = \max_{1 \leq k \leq V} \text{Sim}(S_P, D_{(k,v)})
\]

Identification of the person is estimated as follows.

\[
k^* = \arg\max_{1 \leq k \leq K} \text{Sim}_k(S_P, S_G)
\]