Multiple Faces Tracking Based on Relevance Vector Machine

Wenxing Li and Liming Lian
Electromechnical Engineering College; Xinxiang University,Henan, 451003, China

Abstract— **A multiple faces tracking system was presented based on Relevance Vector Machine (RVM) and Boosting learning. At the first frame, a face detector based on AdaBoost is used to detect faces, and the face motion models and face color models are created. The face motion model consists of a set of RVMs that learn the relationship between the motion of the face and its appearance in the image, and the face color model is the 2D histogram of the face region in CrCb color space. In the tracking process, different tracking methods are used according to different states of the faces and the states are changed according to the tracking results. When the full image search condition is satisfied, a full image search is started in order to find new coming faces and former occluded faces.**

Index Terms— **Multiple faces tracking, relevance vector machine**

I. Introduction

Visual tracking in image sequence is a major research topic in computer vision and it has many potential application areas such as intelligence visual surveillance, human-computer interaction, video coding, etc. Tracking can be considered to be equivalent to establishing coherent relations of image features between frames with respect to position, velocity, shape, texture, color, etc [1]. In the research of face tracking, the common methods are skin-color based tracking [2-4], motion based tracking [5] and feature based tracking [6,7]. Schwerdt *et al.* [2] present a system for tracking single face and controlling camera by panning, tilting, and zooming to maintain the face at the center position. Solar *et al.* [3] use background subtraction and skin detection to track multi-faces. Lerdsudwichai *et al.* [4] use non-parametric distribution to represent the colors of the face region and use mean shift algorithm to track multi-faces. Wang *et al.* [5] present a face tracking system based on a combination of motion detection and template matching. Liang *et al.* [7] build a 3D face model based on the 2D facial features, and estimate the locations of 2D facial features for the next frame using Kalman filters, this system can only track single face.

Recent years, tracking based on statistical learning interests more and more researchers. Avidan [8,9] presents the Support Vector Tracking (SVT) that integrates the Support Vector Machine (SVM) classifier into an optic-flow-based tracker, and tracking is achieved by maximizing the SVM classification score. Williams *et al.* [10,11] use Relevance Vector Machine (RVM) to build a displacement expert which directly estimates displacement from the target region. This algorithm is computation efficiency and very robust, but it can only track one target.

This paper extends the work by Williams *et al.* and presents a multiple faces tracking system based on RVM and Boosting learning. In our system, each face is modeled by a motion model and a color model. The motion model is obtained by training a set of RVMs to learn the relationship between the motion of the face and its appearance in the image, and the color model is obtained by creating the 2D histogram in the CrCb color space. In the tracking process, different tracking methods are used according to the face states, and the states are changed according to the tracking results. Experimental results demonstrate the efficiency of the proposed algorithm.

II. Face Model

In our system, face models are learned when new faces are detected. The face model consists of motion model and color model. The motion model is obtained by training a set of RVMs, and the color model is obtained by creating the 2D histogram in the CrCb color space. In this section, we discuss the RVM briefly first, and then show the learning algorithms of motion model and color model.

A. Relevance Vector Machine

The Relevance Vector Machine (RVM) proposed by Tripping is a model of identical functional form to the popular Support Vector Machine (SVM) [12]. The learning of RVM is under the Bayesian framework, and the RVM yields a probabilistic output. The output of RVM is:

$$y(\mathrm{x};\mathrm{w}) = \sum_{i=1}^{N} w_i k(\mathrm{x},\mathrm{z}_i) + w_0 = \mathrm{w}^{\mathrm{T}}\mathrm{k} \tag{1}$$

Where x is the input vector, $\{\mathrm{z}_i\}$ are the training examples, $\mathrm{w}=[w_0,...,w_N]^{\mathrm{T}}$ is the weights vector, $\mathrm{k}=[1, k(\mathrm{x},\mathrm{z}_1) ,..., k(\mathrm{x},\mathrm{z}_N)]^{\mathrm{T}}$ is the kernel function vector. In RVM, there is no necessity for Mercer kernels and no error/margin trade-off parameter [13].

When RVM is used for regression under the Bayesian framework, a independent zero-mean Gaussian prior distribution is specified over the weights $\{w_i\}$, w_i~$N(0, \alpha_i^{-1})$, so the distribution over the weights vector w is also a Gaussian distribution:

$$p(\mathrm{w}\,|\,\alpha) = \prod_{i=0}^{N} p(w_i\,|\,\alpha_i) = \prod_{i=0}^{N} N(0,\alpha_i^{-1}) \tag{2}$$

Where $\alpha=[\alpha_0,...,\alpha_N]^{\mathrm{T}}$ is a vector of N+1 hyperparameters that should be estimated in the learning process. The training targets are assumed to be sampled

doi:10.4304/jsw.7.4.810-813

with additive noise:

$$t_i = y(\mathrm{z}_i;\mathrm{w}) + \varepsilon_i \tag{3}$$

Where $\{\varepsilon_i\}$ are independent zero-mean Gaussian process noise, $\varepsilon_i \sim N(0,\sigma^2)$, and σ^2 is another hyperparameter that should be estimated in the learning process. Obviously, the distribution over the targets vector $\mathrm{t}=[t_0,...,t_N]^{\mathrm{T}}$ is also a Gaussian distribution:

$$p(\mathrm{t} \mid \mathrm{w},\sigma^2) = \prod_{i=1}^{N} p(t_i \mid \mathrm{w},\sigma^2) = \prod_{i=1}^{N} N(y(\mathrm{z}_i;\mathrm{w}),\sigma^2) \tag{4}$$

The posterior distribution over the weights is thus given by:

$$p(\mathrm{w} \mid \mathrm{t},\alpha,\sigma^2) = \frac{p(\mathrm{t} \mid \mathrm{w},\sigma^2)p(\mathrm{w} \mid \alpha)}{\int p(\mathrm{t} \mid \mathrm{w},\sigma^2)p(\mathrm{w} \mid \alpha)d\,\mathrm{w}} = N(\boldsymbol{\mu},\Sigma) \tag{5}$$

Where $\mu=\sigma^{-2}\Sigma\Phi^{\mathrm{T}}\mathrm{t}$, $\Sigma=(\sigma^{-2}\Phi^{\mathrm{T}}\Phi+\mathrm{A})^{-1}$, $\mathrm{A}=\mathrm{diag}(\alpha_0,...,\alpha_N)$, $\Phi=[\Phi(\mathrm{z}_1),...,\Phi(\mathrm{z}_\mathrm{N})]^{\mathrm{T}}$, $\Phi(\mathrm{z}_i)=[1,k(\mathrm{z}_i,\mathrm{z}_1),..., k(\mathrm{z}_i,\mathrm{z}_N)]^{\mathrm{T}}$. Then, the parameters w can be set to fixed values μ for the purpose of prediction. The values of μ is dependent on the hyperparameters α and σ^2, and the values of α and σ^2 can be obtained by maximize the marginal likelihood $p(\mathrm{t}|\alpha, \sigma^2)$. In this paper, we use the fast marginal likelihood maximization method presented in Ref. [14][15]. For the full details of RVM and the fast maximization method, one can peruse Ref. [12].

B. Face Motion Model

The face motion model consists of a set of RVMs, and each RVM is trained to learn the relationship between one motion of a face and its appearance in the image [10]. In this paper, the motion model has three RVMs: $y_\mathrm{H}(\mathrm{x};\mathrm{w}_\mathrm{H})$, $y_\mathrm{V}(\mathrm{x};\mathrm{w}_\mathrm{V})$ and $y_\mathrm{S}(\mathrm{x};\mathrm{w}_\mathrm{S})$, corresponding to horizontal translation t_H, vertical translation t_V and scale change t_S, respectively. The ranges of the three motions are: $t_\mathrm{H}\in[-\Delta_\mathrm{H}, \Delta_\mathrm{H}]$, $t_\mathrm{V}\in[-\Delta_\mathrm{V}, \Delta_\mathrm{V}]$, $t_\mathrm{S}\in[-\Delta_\mathrm{S}, \Delta_\mathrm{S}]$. Algorithm 1 shows the learning algorithm of the face motion model.

```
Algorithm 1: Learn face motion model
0. (Input)
      (1) Current image, $I$;
      (2) Face region, $\lambda$;
      (3) Number of examples, $N$;
      (4) Motion ranges, $\Delta_\mathrm{H}$, $\Delta_\mathrm{V}$ and $\Delta_\mathrm{S}$;
1. (Sampling)
      For $i$=1,...,$N$:
        (1) $t_\mathrm{H}^i \leftarrow$Uniform[$-\Delta_\mathrm{H}$, $\Delta_\mathrm{H}$], $t_\mathrm{V}^i \leftarrow$Uniform[$-\Delta_\mathrm{V}$, $\Delta_\mathrm{V}$],
            $t_\mathrm{S}^i \leftarrow$Uniform[$-\Delta_\mathrm{S}$, $\Delta_\mathrm{S}$];
        (2) $img \leftarrow$Sample($I$, $\lambda$, $t_\mathrm{H}^i$, $t_\mathrm{V}^i$, $t_\mathrm{S}^i$), move $\lambda$ horizontally,
            vertically and make scale change in $I$ using $t_\mathrm{H}^i$, $t_\mathrm{V}^i$
            and $t_\mathrm{S}^i$, then sample $\lambda$ from $I$ into a patch $img$;
        (3) Change $img$ into gray-scale image and do
histogram equalization;
        (4) Raster scan $img$ into a vector $\mathrm{z}_i$;
2. (Tranining)
      $\mathrm{w}_\mathrm{H}$, $\Sigma_\mathrm{H} \leftarrow$RVMTrain($\{\mathrm{z}_i\}$,$\{t_\mathrm{H}^i\}$), $\mathrm{w}_\mathrm{V}$,
   $\Sigma_\mathrm{V} \leftarrow$RVMTrain($\{\mathrm{z}_i\}$,$\{t_\mathrm{V}^i\}$),
   $\mathrm{w}_\mathrm{S}$, $\Sigma_\mathrm{S} \leftarrow$RVMTrain($\{\mathrm{z}_i\}$,$\{t_\mathrm{S}^i\}$);
3. (Output)
      $\mathrm{w}_\mathrm{H}$, $\Sigma_\mathrm{H}$; $\mathrm{w}_\mathrm{V}$, $\Sigma_\mathrm{V}$; $\mathrm{w}_\mathrm{S}$, $\Sigma_\mathrm{S}$.
```

C. Face Color Model

The color model records the skin distribution of the face region, and we use the lighting compensation method in Ref. [16] to reduce the color sensitivity to lighting variations. After lighting compensation, the image is translated into the YCrCb color space and the 2D color histogram in CrCb space is used as the face color model. The learning algorithm of face color model is show in Algorithm 2.

```
Algorithm 2: Learn face color model
0. (Input)
      (1) Current image, $I$;
      (2) Face region, $\lambda$;
1. (Lighting compensation)
      Select region $\lambda$ from $I$ and make lighting compensation for
      region $\lambda$;
2. (2D histogram)
      Translate $\lambda$ into YCrCb space and calculate the 2D
      histogram $H_\mathrm{model}^\mathrm{CrCb}$ in CrCb space;
3. (Output)
      The 2D histogram, $H_\mathrm{model}^\mathrm{CrCb}$.
```

III. Face Tracking

In the face tracking process, the system use motion model to track the face, and the tracking result will be validated. If the validator gives a negative response, it will trigger a local search in the next frame.

A. Face Tracking Based on Motion Model

The position of the face in current frame should be predicted using the motion model. Denote the face region at time instance t by u_t, and assume that we have known the face region u_{t-1}, so the displacement of u_{t-1} from the true face region u_t can be estimated as $t_\mathrm{H}=\mathrm{w}_\mathrm{H}^\mathrm{T}\mathrm{k}$, $t_\mathrm{V}=\mathrm{w}_\mathrm{V}^\mathrm{T}\mathrm{k}$, $t_\mathrm{S}=\mathrm{w}_\mathrm{S}^\mathrm{T}\mathrm{k}$, and the variance is $s_\mathrm{H}=\mathrm{k}^\mathrm{T}\Sigma_\mathrm{H}\mathrm{k}+\sigma_\mathrm{H}^2$, $s_\mathrm{V}=\mathrm{k}^\mathrm{T}\Sigma_\mathrm{V}\mathrm{k}+\sigma_\mathrm{V}^2$, $s_\mathrm{S}=\mathrm{k}^\mathrm{T}\Sigma_\mathrm{S}\mathrm{k}+\sigma_\mathrm{S}^2$. Let $\mathrm{t}=[t_\mathrm{H}, t_\mathrm{V}, t_\mathrm{S}]^\mathrm{T}$, and $\mathrm{t}\sim N(W\mathrm{k}, S)$, where $W=[\mathrm{w}_\mathrm{H}, \mathrm{w}_\mathrm{V}, \mathrm{w}_\mathrm{S}]^\mathrm{T}$, $S=\mathrm{diag}(s_\mathrm{H}, s_\mathrm{V}, s_\mathrm{S})$. Thus u_t can be calculated using Kalman Filter [10], and the Kalman state formulation is:

$$\mathrm{u}_t = F\mathrm{u}_{t-1} + v \qquad v\sim N(0, Q) \tag{6}$$

Where F and Q can be learned from a hand-labeling motion sequence. The face tracking algorithm based on motion model is shown in Algorithm 3.

```
Algorithm 3: Face tracking based on motion model
(RVM tracking)
0. (Input)
      (1) Current image, $I$;
      (2) Face region at time instance $t$-1, $\mathrm{u}_{t-1}$;
      (3) State covariance, $U_{t-1}$;
1. (Prediction)
      $\mathrm{u}_t \leftarrow F\mathrm{u}_{t-1}$, $U_t \leftarrow FU_{t-1}F^\mathrm{T} + Q$;
2. (RVM regression)
      (1) $img \leftarrow$Sample($I$, $\mathrm{u}_{t-1}$), sample $\mathrm{u}_{t-1}$ from $I$ into a patch
$img$;
      (3) Change $img$ into gray-scale image and do histogram
equalization;
      (4) Raster scan $img$ into a vector x;
```

```
        (2) Calculate t and $S$ using $y_H(x; w_H)$, $y_V(x; w_V)$ and $y_S(x; w_S)$;
3. (Kalman gain)
        $G= U_t [U_t +S]^{-1}$;
4. (Correction)
        $u_t \leftarrow u_t + Gt$, $U_t \leftarrow U_t - GU_t$;
3. (Output)
        $u_t$, $U_t$.
```

B. Tracking Result Validation

The tracking result calculated by Algorithm 3 should be validated and the face state will be reset according to the validation result. For a successful tracking, on one hand the tracking result should contain face; on the other hand the face should be the same as the one be tracked in last frame. Here we use the face detector based on AdaBoost [17] and the face color model as the validator. The tracking result validation is shown in Algorithm 4.

```
Algorithm 4: Tracking result validation
0. (Input)
        (1) Current image, $I$;
        (2) Tracking result, $u_t$;
        (3) Similarity threshold, $threshold\_1$;
1. (Validation using face detector)
        (1) Detect faces in $u_t$ using face detector based on FloatBoost;
        (2) **If** $u_t$ contains face, then $test1 \leftarrow$PASS;
            **Else** $test1 \leftarrow$not PASS;
2. (Validation using color model)
        (1) Make lighting compensation for region $u_t$;
        (2) Translate $u_t$ into YCrCb color space and calculate the
            2D histogram $H_u^{CrCb}$ in CrCb space;
        (3) Calculate the similarity, $s$, between $H_u^{CrCb}$ and $H_{model}^{CrCb}$:
$$s = \frac{\sum_m \sum_n [H_u^{CrCb}(m,n) - \overline{H_u^{CrCb}}][H_{model}^{CrCb}(m,n) - \overline{H_{model}^{CrCb}}]}{\sqrt{\sum_m \sum_n [H_u^{CrCb}(m,n) - \overline{H_u^{CrCb}}]^2 \sum_m \sum_n [H_{model}^{CrCb}(m,n) - \overline{H_{model}^{CrCb}}]^2}};$$
        (4) **If** $s > threshold\_1$, **then** $test2 \leftarrow$PASS;
            **Else** $test2 \leftarrow$not PASS;
3. (Output)
        **If** $test1$=PASS and $test1$=PASS, **then** tracking successful;
        **Else** tracking failed.
```

IV. Face Matching

In our system, the full image search is started when the full or local image search condition is satisfied. In these two kinds of search, the face detector based on AdaBoost is employed. It may find multiple faces in these two kinds of search, so we need to match the detected faces to the known faces recorded in the face models database.

A. Similarity Matrix

The face detector may find multiple faces in full image search or local search, and we should match the detected faces to the known faces in the face models database.

For the purpose of matching faces efficiently with the three constraints above, we introduce the similarity matrix. Similarity matrix, C, is a $m \times n$ matrix, where m is the number of detected faces and n is the number of faces in the face models database. The element c_{ij} in C is the similarity between detected face i and known face j in the face models database.

B. Face Matching Based on Similarity Matrix

Similarity matrix C reflects the similarity between detected faces and known faces in the face models database with the constrains of the face states. In full image search, we can use C to find whether the faces reappear and whether new faces appear in the scene. In local search, we can use C to find whether the face can be found in its local search area. Algorithm 5 is the face matching algorithms in full or loacl image search.

```
Algorithm 5: Face matching based on similarity matrix
0. (Input)
        (1) Number of detected faces, $m$;
        (4) Similarity matrix, $C$;
        (5) Similarity threshold, $t$;
1. (Match face)
        (1) $max \leftarrow$the max value in the column of $C$,  $r \leftarrow$the row index of $max$;
        (3) **While** $max > t$;
                **If** $max$ is the max value in row $r$ of $C$, **then** the face
                    is mathced;
                **Else** (a) $max \leftarrow$the max value that is less than $max$ in the column of $C$;
                         (b) $r \leftarrow$the row index of $max$;
```

V. Experimental Results

To demonstrate the effectiveness of our system, an image sequences captured by a digital camera are used to test our system. The sequence has a resolution of 320×240, 24bit color. The algorithm runs on a 2.4GHz PC without code optimization.

A. Parameter settings

In this paper, Gaussian kernel function is used:

$$k(z_i, z_j) = \exp(-\frac{1}{2Mk^2}\|z_i - z_j\|^2) \tag{7}$$

Where M is the input dimensionality and k controls the width of $k(,)$.

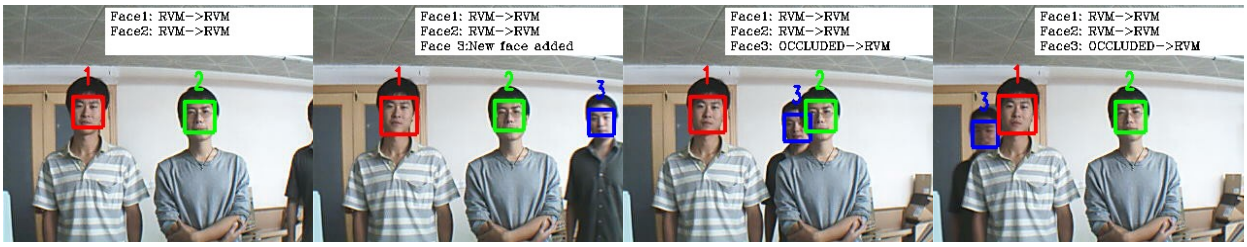

Fig.1. Some tracking results of the image sequence (frame 1, 80, 125, 216);

B. Results

Fig.1 shows some tracking results of the image sequence. In the results, the faces are shown in rectangles, and the number above the rectangle is the face ID. There are 2 persons (Face 1 and Face 2) in image sequences 1 first, then the third person (Face 3) come in the scene and is occluded by Face 1 and Face 2 in turn, then Face 3 leaves the scene for a while before it returns and occluded by Face 1 and Face 2 again. The results show that Face 3 can be detected as a new face when he come into the scene, and can be tracked correctly after he was occluded by others.

VI. Conclusion

This paper presents a multiple faces tracking system based on RVM and AdaBoost. At the start of the system, a face detector based on AdaBoost is used to detect faces and the face models are created. The face model in our system include the motion model that consists of a set Relevance Vector Machines and the color model that is the 2D histogram in CrCb color space. In the tracking process, different tracking methods are used according to the different states of the face, and a full image search is started at the regular intervals of several frames in order to find new coming faces and occluded faces. In the full image search and local search, the similarity matrix is introduced to help matching faces efficiently. Experimental results demonstrate that our system can detect new faces automatically and handle the problems of faces occlusion and face scale change.

Reference

[1] Hu, W.M., Tan, T.N., Wang, L., Maybank, S.J.: A survey on visual surveillance of object motion and behaviors. IEEE Trans. on System Man and Cybernetics 34 (2004) 334-351

[2] Schwerdt, K., Crowley, J.: Robust face tracking using color. In: Proc. of Automatic Face and Gesture Recognition, Grenoble, (2000) 90–95

[3] Solar, J.R., Shats, A., Verschae, R.: Real-time tracking of multiple persons. In: Proc. of the 12th International Conference on Image Analysis and Processing (2003)

[4] Lerdsudwichai, C., Abdel-Mottaleb, M., Ansari, A.N.: Tracking multiple people with recovery from partial and total occlusion. Pattern Recognition 38 (2005) 1059-1070

[5] Wang, L., Tan, T.N., Hu, W.M.: Face tracking using motion-guided dynamic template matching. In: Fifth Asian Conference on Computer Vision (2002)

[6] Colmenarez, A., Lopez, R., Huang, T.: 3D model-based head tracking. In: Proc. of Visual Communications and Image Processing (1997)

[7] Liang, R., Chen, C., Bu, J.: Real-time facial features tracker with motion estimation and feedback. In: Proc. of the International Conference on Systems, Man and Cybernetics (2003) 3744–3749

[8] Avidan, S.: Support vector tracking. IEEE Trans. On Pattern Analysis and Machine Intelligence 26 (2004) 1064-1072

[9] Avidan, S.: Support vector tracking. In: Proc. of International Conference on Computer Vision and Pattern Recognition (2001) vol. 1: 184-192

[10] Williams, O., Blake, A., Cipolla, R.: Sparse Bayesian learning for efficient visual tracking. IEEE Trans. On Pattern Analysis and Machine Intelligence 27 (2005) 1292-1304

[11] Williams, O., Blake, A., Cipolla, R.: A sparse probabilistic learning algorithm for real-time tracking. In: Proc. of the Ninth International Conference on Computer Vision (2003)

[12] Tipping, M.E.: Sparse Bayesian learning and relevance vector machine. Journal of Machine Learning Research 1 (2002) 211-244

[13] Burges, C.J.C.: A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery 2 (1998) 121-167

[14] Faul, A.C., Tipping, M.E.: Analysis of sparse Bayesian learning. In: Dietterich, T.G., Becker, S., Ghahramani, Z., ed.: Advances in Neural Information Processing Systems (2002) 383–389

[15] Tipping, M.E., Faul, A.C.: Fast marginal likelihood maximisation for sparse Bayesian models. In: Bishop, C.M., Frey, B.J., ed: Proc. of the Ninth International Workshop on Artificial Intelligence and Statistics (2003)

[16] Hsu, R.L.: Abdel-Mottaleb M. Face detection in color images. IEEE Trans. On Pattern Analysis and Machine Intelligence 24 (2002) 696-706

[17] Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proc. of International Conference on Computer Vision and Pattern Recognition (2001) vol. 1: 511-518