AAM based Face Tracking with Temporal Matching and Face Segmentation

Mingcai Zhou[1*] Lin Liang[2] Jian Sun[2] Yangsheng Wang[1]
[1]Institute of Automation
Chinese Academy of Sciences, Beijing, China
{mingcai.zhou, yangsheng.wang}@ia.ac.cn
[2]Microsoft Research Asia
Beijing, China
{lliang, jiansun}@microsoft.com

Abstract

Active Appearance Model (AAM) based face tracking has advantages of accurate alignment, high efficiency, and effectiveness for handling face deformation. However, AAM suffers from the generalization problem and has difficulties in images with cluttered backgrounds. In this paper, we introduce two novel constraints into AAM fitting to address the above problems. We first introduce a temporal matching constraint in AAM fitting. In the proposed fitting scheme, the temporal matching enforces an inter-frame local appearance constraint between frames. The resulting model takes advantage of temporal matching's good generalizability, but does not suffer from the mismatched points. To make AAM more stable for cluttered backgrounds, we introduce a color-based face segmentation as a soft constraint. Both constraints effectively improve the AAM tracker's performance, as demonstrated with experiments on various challenging real-world videos.

1. Introduction

Face tracking is useful for many applications, such as video conferencing, gaming, surveillance, facial expression analysis, and animated avatars for web communication. A good face tracker should work stably and accurately for different illuminations and environments, persons, head motions and face expressions, and run continuously for long sequences without drift or error accumulation.

There are two main categories of face tracking algorithms. The first category is feature-based tracking, which matches the local interest-points between subsequent frames to update the tracking parameters, such as a 3D pose tracker [19, 20] and 3D deformable face tracking [22]. Because local feature matching does not depend on the training data, the feature-based tracking is less sensitive to variation in illumination and object appearance. Furthermore, the coarse-to-fine local feature search scheme [20] can effectively handle fast motion. One limitation of this approach is that the feature matching is error-prone resulting in jittery and inaccurate tracking.

The second category is appearance-based, using generative linear models of face appearance, such as 2D Active Appearance Models (AAM) [6] and 3D Morphable Models [3]. Compared to the feature-based tracking, AAM can track a face more accurately and stably with little jitter. However, AAM may have difficulty generalizing to unseen images because AAM is trained from a set of example faces. Also AAM is sensitive to the initial shape and may easily be stuck in local minima because of its gradient decent optimization. Cluttered backgrounds also reduce AAM stability in tracking a face outline.

In this work, we introduce two novel constraints to make AAM fitting more robust. We first incorporate a temporal matching constraint into AAM fitting. The temporal matching constraint is an inter-frame local appearance constraint between successive frames. The face shape is optimized by considering not only the AAM model fitting error, but also inter-frame local matching error. The resulting AAM tracker is more general because the introduced temporal matching term is not related to the training data. At the same time, our tracker does not suffer from mismatched points because we search the matching points during AAM fitting instead of directly constraining the points at the prematched positions. Furthermore, we initialize the shape based on the correspondences found by a robust local feature matching. The resulting initial shape is closer to the ground-truth shape, hence it improves the stability in tracking fast face motions.

To track the face outline more stably in cluttered backgrounds, we introduce a color-based face segmentation as an additional constraint in AAM fitting. The key to our technique is that we incorporate the face segmentation as a soft constraint, which works with inaccurate segmentation.

*This work was done when Mingcai Zhou was visiting at Microsoft Research Asia

2. Related Work

Currently there are two main approaches to incorporate a temporal constraint for appearance-based tracking. One is feature-based approach that matches the local features between subsequent frames and directly constrains the feature points at the matched positions, such as Liao's work[11]. A feature-based approach may suffer from mismatched local features. The other approach is intensity-based; it matches the global image appearances between subsequent frames, as in Liu's work [12]. The intensity-based approach expects brightness consistency between the images, and it is sensitive to fast illumination changes. Our method combines the strengths of both approaches while mitigating their limitations.

The Constrained Local Model (CLM) developed by Cristinacce and Cootes [9] represents a face as a combination of shape and local feature templates. The model is fitted to the image by optimizing the shape parameters to match the image's local appearances to the templates. Our AAM tracker also enjoys the benefits of such local feature matching by incorporating an inter-frame local appearance constraint.

Many works have adopted color-based face segmentation to help face tracking [17][1][5]. Most of them use face segmentation to initialize the face location but not as a constraint during the online fitting. Choi's work [5] fits an AAM to a preprocessed image containing only the segmented face region. This approach relies on a very accurate segmentation, whereas ours does not.

3. AAM based Face Tracking

3.1. Active Appearance Models

Assuming that a shape $\mathbf{s}$ is described by N feature points, $\mathbf{s} = [x_1, y_1, x_2, y_2, \ldots, x_N, y_N]$ in the image, a shape is represented in AAM as a mean shape $\mathbf{s}_0$ plus a linear combination of n shape bases $\{\mathbf{s}_i\}$:

$$\mathbf{s}(\mathbf{p}) = \mathbf{s}_0 + \sum_{i=1}^{n} p_i \mathbf{s}_i, \tag{1}$$

where $\mathbf{p} = [p_1, p_2, \ldots, p_n]$ are the shape parameters. Usually, the mean shape $\mathbf{s}_0$ and the shape bases $\{\mathbf{s}_i\}$ are learned by applying PCA to the training shapes, Figure 1 (a) shows some examples of the shape bases. To consider global transformation of a shape, the shape bases set $\{\mathbf{s}_i\}$ is expanded to include four additional bases representing global translation, scaling, and rotation [14].

The appearance A of the AAM is defined as the image patch enclosed by the mean shape $\mathbf{s}_0$. Similar to shape, the appearance A is represented as a mean appearance A_0 plus a linear combination of m appearance bases $\{A_i\}$:

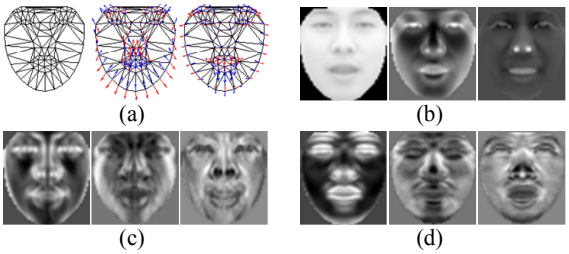


Figure 1. Multi-band AAMs. (a) Mean shape $\mathbf{s}_0$ and the first two shape bases learned by PCA. (b) to (d) are mean appearance and the first two appearance bases of the intensity band, x-direction band and y-direction band respectively.

$$A = A_0 + \sum_{i=1}^{m} \lambda_i A_i, \tag{2}$$

where the coefficients $\{\lambda_i\}$ are the appearance parameters. The mean appearance A_0 and appearance bases $\{A_i\}$ are learned by applying PCA to the shape-normalized training images [6], Figure 1 (b) shows some examples of the appearance bases.

To locate the shape on an observed image I, AAM aims to find the optimal shape parameters $\mathbf{p}$ and appearance parameters λ to minimize the difference between the warped-back appearance $I(\mathbf{W}(\mathbf{p}))$ and the synthesized appearance A_λ:

$$\begin{aligned} E_a(\mathbf{p}, \lambda) &= \|A_\lambda - I(\mathbf{W}(\mathbf{p}))\|_2 \\ &= \sum_{\mathbf{x} \in \mathbf{s}_0} [A_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i A_i(\mathbf{x}) - I(\mathbf{W}(\mathbf{x}; \mathbf{p}))]^2, \end{aligned} \tag{3}$$

where $\mathbf{W}(\mathbf{x}; \mathbf{p})$ is a warping function defined to map every pixel $\mathbf{x}$ in the model coordinate to its corresponding image point. Usually $\mathbf{W}(\mathbf{x}; \mathbf{p})$ is a piecewise affine warp [14] defined by the pair of shapes $\mathbf{s}_0$ and $\mathbf{s}(\mathbf{p})$: for each triangle in $\mathbf{s}_0$ there is a corresponding triangle in $\mathbf{s}(\mathbf{p})$ and each pair of triangles defines an affine warp.

The cost function (3) can be efficiently minimized by the inverse compositional parameter update technique [14].

3.2. Basic AAM-based Face tracking

In our system, we extend the basic AAM technique in the following aspects to make the tracking more stable.

Multi-Band AAMs with Edge Structure. We adopt the multi-band appearance model [23] to improve the tracker's generalizability. In our system, the appearance is a concatenation of three texture band values: the intensity, x-direction gradient strength, and y-direction gradient strength. Figure 1 shows the leading shape and appearance bases of our multi-band AAMs.

Figure 2. View-based AAMs. Here just shows the mean appearances of the intensity band for each view.

View-Based AAMs. To handle large angles of head rotation, we adopt a view-based approach [8]. Three AAMs are trained for view ranges $[-60°, -30°]$, $[-30°, 30°]$ and $[30°, 60°]$ respectively. Figure 2 shows the mean appearances of the intensity band for each view.

Tracking Initialization and Recovery. We consider the tracking lost if the appearance reconstruction error (Equation 3) exceeds a pre-defined threshold. To initialize the tracking, we first do face detection [21] followed by component detection. The facial components (the eyes, brows, nose and mouth) are constrained [7] during AAM.

4. Temporal Matching Constrained AAM

In this section, we introduce the formulation of our temporal matching constraint and propose a robust shape initialization method based on face motion direction.

4.1. Temporal Matching Constraint

We expect to improve AAM's generalizability by considering the inter-frame correspondences. One possible way is to constrain the global appearance consistency between subsequent frames. But such an approach is sensitive to fast illumination or texture changes. Another way is to do feature matching between frames and then constrain the feature points at the pre-matched positions during AAM fitting. The problem with this approach is that the mismatched points may lead to jitter and inaccurate tracking results. To address these issues, we propose a temporal matching approach that constrains the inter-frame local appearance consistency during AAM fitting.

We select some feature points with salient local appearances at the previous frame and optimize the shape parameters to match the local appearances of the selected feature points between current and previous frames. Our temporal matching approach is illustrated in Figure 3. We warp the previous frame I_{t-1} to the model coordinate and get the appearance A_{t-1}. Using the j-th feature point's local match R^j_{t-1} as an example, the warping function $\mathbf{W}(\mathbf{x};\mathbf{p}_t)$ maps R^j_{t-1} to a patch R^j_t at frame t using the current shape parameters $\mathbf{p}_t$. We optimize the shape parameters $\mathbf{p}_t$ to change local patch R^j_t's position and rotation such that the appearance of R^j_t matches R^j_{t-1}.

Mathematically, we formulate the temporal matching error between I_{t-1} and I_t as a function of the shape parameters $\mathbf{p}_t$:

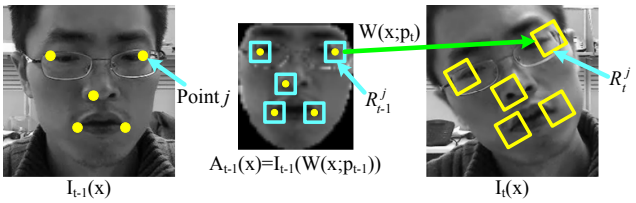


Figure 3. Temporal matching constraint. Some feature points are selected at the frame t-1. For clarity here, we only show five points for illustration. A_{t-1} is the appearance of frame I_{t-1} warped to the model's coordinate. Given current shape parameters $\mathbf{p}_t$, the warping function $\mathbf{W}(\mathbf{x};\mathbf{p}_t)$ maps the patches in A_{t-1} to I_t, for example, the j-th patch R^j_{t-1} is mapped to R^j_t. We adjust the shape parameters $\mathbf{p}_t$ to match the local patches appearance in I_t to those in A_{t-1}.

$$E_t = \sum_{j\in\Omega_t}\sum_{\mathbf{x}\in R^j}[A_{t-1}(\mathbf{x})/\bar{g}^j_{t-1} - I_t(W(\mathbf{x};\mathbf{p}_t))/\bar{g}^j_t(\mathbf{p}_t)]^2, \tag{4}$$

where Ω_t is a set of feature points, including some interesting points selected by a corner detector [18] and some semantic points, such as the eyes' corners. A_{t-1} is the face appearance of frame $t-1$ in the model coordinate, R^j is the local patch corresponding to the j-th feature point. In our experiments, the size of R^j is 9×9, $\bar{g}^j_{t-1}$ and $\bar{g}^j_t(\mathbf{p}_t)$ are the average intensity of the j-th patches of frame $t-1$ and t respectively. $\bar{g}^j_{t-1}$ and $\bar{g}^j_t(\mathbf{p}_t)$ are used to normalize the illuminations of two patches. Next we add the cost (4) as a new term to the AAM cost function (3):

$$E = E_a + w_t E_t, \tag{5}$$

where w_t controls the strength of the temporal matching constraint. Empirically, we found that $w_t = 100$ is appropriate for most cases.

With the temporal matching term, our tracker has improved generalizability. Since we match the local patches, our tracker is resistant to global illumination changes. Furthermore, our tracker does not suffer from the mismatched points, since feature matching is continuously refined by updating the shape parameters during AAM fitting.

The cost (5) can still be efficiently minimized based on inverse compositional algorithm [2]. Notice that the patch's average intensity $\bar{g}^j_t(\mathbf{p}_t)$ is a function of the shape parameter $\mathbf{p}_t$, which complicates the optimization. So to simplify the optimization we approximate $\bar{g}^j_t(\mathbf{p}_t)$ using the patches sampled from the estimated locations of the previous iteration and fix $\bar{g}^j_t(\mathbf{p}_t)$ during the following steps.

Good initial parameters are crucial to the success of AAM fitting. To initialize the shape, we develop a robust estimation algorithm based on the face motion direction, as explained in the following section.

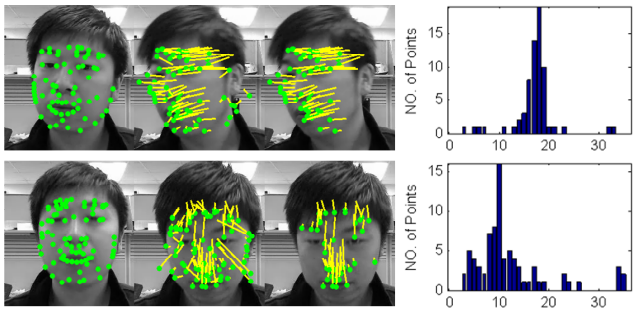


Figure 4. Reject the matching outliers by main direction filter. (a) Selected feature points at frame $t-1$. (b) Matched feature points at frame t. The lines show the moving directions of the points. (c) Remaining feature points after main direction filter. (d) The histogram of the motion angle. Here we discretize the angle's range $[0, 2\pi]$ to 36 bins and add an additional bin for the fixed points.

4.2. Robust Shape Initialization Based on Face Motion Direction

At frame t, to initialize the shape parameters $\mathbf{p}_t$, as shown in Figure 5 (a), we search locally to find the matching feature points of frame $t-1$, and then reject the outliers by the main direction of the face motion. Finally, the shape parameters $\mathbf{p}_t$ are estimated to fit to the remained points.

As shown in Figure 4, there usually exists a main motion direction for facial motions. Those feature points whose motion directions are inconsistent with the main direction are most likely to be outliers. Denoting θ as the angle of motion for a feature point and $\theta \in [0, 2\pi]$, we estimate the main motion direction based on the angle θ's histogram, as shown in Figure 4 (d). The main motion's angle Θ corresponds to the bin with maximum points. Given a threshold $\Delta\Theta$, we filter out those points whose directions are not consistent with the main direction: $\theta \ni [\Theta - \Delta\Theta, \Theta + \Delta\Theta]$. In our experiments, $\Delta\Theta$ is set as $\pi/9$. Figure 4 (c) shows an example filtered by the main direction. Mismatched points are rejected effectively. Compared with RANSAC [13], our motion direction filter is more efficient with comparable results.

Suppose, after filtering by the main motion direction, M matched points $\{\mathbf{z}_i\}_{i=1}^{M}$ remain. Then we estimate the initial shape parameters $\mathbf{p}_0$ to fit these points; that is, minimize the following cost function:

$$\mathbf{p}_0 = \min_{\mathbf{p}} \sum_{i=1}^{M} -w_i \rho(\| \sum_{j=1}^{3} c_{ij} \mathbf{W}(\mathbf{x}_{ij}; \mathbf{p}) - \mathbf{z}_i \|, r), \quad (6)$$

where the weight $w_i = \cos(\theta_i - \Theta)$ represents the consistency of feature point i's direction with the main motion direction. The function $\rho(\cdot, r)$ is a m-estimator as in [16], and $\sum_{j=1}^{3} c_{ij} \mathbf{W}(\mathbf{x}_{ij}; \mathbf{p})$ is the estimated position of the point i given the shape parameters $\mathbf{p}$. The vertex coordinates of the triangle containing the point i in model coordinates are $\{\mathbf{x}_{ij}\}_{j=1}^{3}$. The triangle coordinates of the point i are $\{c_{ij}\}_{j=1}^{3}$. We find the optimal $\mathbf{p}_0$ by a Gauss-Newton algorithm [15].

Figure 5 compares the results of different initialization methods. Notice the main direction filter generates a much better initial shape, which encourages the correct alignment.

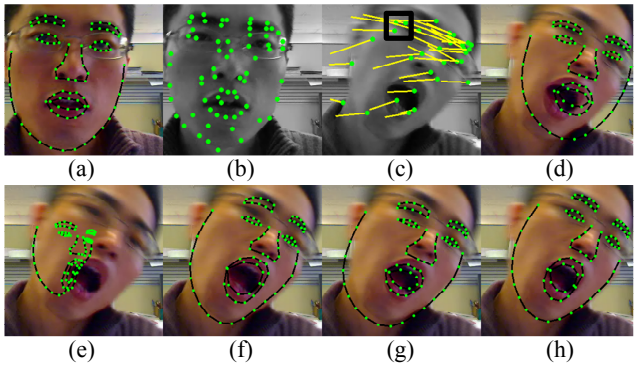


Figure 6. Comparisons of different temporal constraints. The first row shows the process of shape initialization for methods (f) to (h). (a) The shape of frame $t-1$. (b) Selected feature points at frame $t-1$. (c) Feature points used for estimating the initial shape. (d) The initial shape of frame t. The second row shows the results of four methods. (e) Multi-pyramid based AAM fitting. (f) Liao's method [11]. (g) Liu's method [12]. (h) Our method.

4.3. Comparison

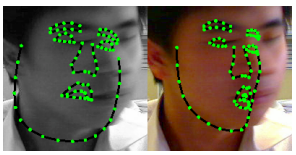
To demonstrate the effectiveness of our proposed temporal matching constraint, we compare our algorithm with three methods. The first is multi-pyramid based AAM fitting [6] using the previous frame's shape as the initial shape without any constraints. The second method is Liao's work [11], which constrains the matched points' positions. The third is Liu's work [12], which matches the global face appearances. These last two methods adopt our technique to initialize the shape (as explained in section 4.2) but use different temporal constraints. As shown in Figure 6, multi-pyramid based method fails for a fast motion because of the poor initial shape. For Liao's approach, the mismatched points (marked by the square in Figure 6 (c)) damage the alignment. Liu's method is sensitive to large appearance changes caused by motion blur, while our approach is more stable.

5. Face Segmentation Constrained AAM

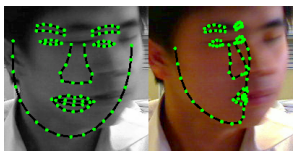
For videos with cluttered backgrounds, AAM tends to fit the face outline to the background edges. This problem is more obvious for AAM using an edge structure. As shown at the first row of Figure 7, although the head keeps still, the face outline moves to the nearby background edge slowly.

To handle this background edge problem, we segment

(a) Shape initialization with the main direction filter. The images from left to right are: the shape of frame $t-1$, the selected feature points at frame $t-1$, the matched feature points at frame t, the remaining feature points after the main direction filter, the initial shape of frame t, and the resulting shape of frame t.



(b) Shape initialization using feature matching without the main direction filter. The left is the initial shape estimated from all the matched feature points (as shown in the third image of (a)). The right is the result.



(c) Shape initialization using previous frame's shape as the initial shape. The left is the initial shape. The right is the result.

Figure 5. Comparisons of different shape initialization methods.

Figure 7. Comparisons of different segmentation constraints. Here we see the tracking results of three sequential frames. The first row shows the results without the segmentation constraint. The second row shows the results of Choi's work [5]. The third row shows the results of our approach.

the face region using an adaptive color model and constrain AAM fitting to encourage the outline points to be located inside the segmented face region. As shown in Figure 8, we transfer the segmentation mask into a cost map I_D. In the map I_D, the pixel values are set to zero inside the face region and increase gradually when moving away from the boundary. Then we add a constraint term E_c into Equation (3) to force the values of the outline points on the cost map I_D to be as small as possible, so as to locate inside the face region. Mathematically, E_c has the form:

$$E_c = \sum_{k=1}^{K} I_D(W(\mathbf{x}_k; \mathbf{p}))^2, \tag{7}$$

(a) (b) (c)

Figure 8. Transfer face segmentation result to a cost map. (a) Segmented face region. (b) The face region mask. (c) The cost map.

where $\{\mathbf{x}_k\}$ are the locations of the selected outline points in the model coordinate.

Now the cost function to minimize becomes:

$$E = E_a + w_t E_t + w_c E_c, \tag{8}$$

where the weight w_c controls the strength of the face segmentation constraint. In our experiment, w_c is set to 0.01. The cost function (8) can still be efficiently minimized based on the inverse compositional technique.

As shown at the third row in Figure 7, with the face segmentation constraint, the tracking is more stable and accurate. The second row shows the results of Choi's work [5], which fits AAM to an image only containing the segmented face region. Choi's approach tends to fail when the segmentation is inaccurate. In contrast, our approach does not change the original image and the constraint is described by the cost map, thus even with an inaccurate segmentation, as shown in Figure 7, the appearance term E_a in Equation (8) can still work to achieve correct alignment.

Face Segmentation. At the first frame, to segment the face after AAM fitting, we calculate the color distributions p_f and p_b of the face and background respectively. p_f and p_b are modeled as mixture Gaussian fitted by K-mean algorithm [10]. To segment the face region, a pixel $\mathbf{x}$ is labeled

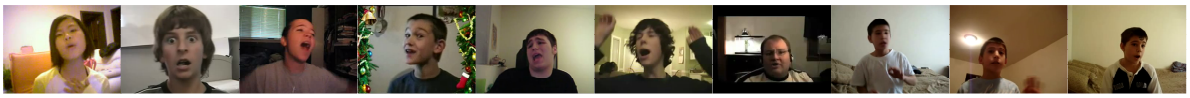


Figure 9. Key frames of test videos.

	v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	sum
Number of Frames	2680	3010	6269	6308	6295	3926	3182	5922	5458	7200	50250
basic AAM	229	44	306	115	73	252	353	354	81	291	2098
AAM+RI	40	20	56	57	32	39	141	75	15	52	527
AAM+RI+TP	24	15	35	30	31	24	112	51	12	25	359
AAM+RI+TA	33	18	49	33	31	16	119	65	14	44	422
AAM+RI+TO	18	16	25	16	23	14	88	36	9	22	267
AAM+RI+TO+FS	11	15	19	15	19	8	76	30	8	17	218

Table 1. Stability comparisons of AAM based tracking algorithms using different shape initialization methods and constraints. v1 to v10 corresponds to the videos shown at Figure 9 from left to right. The second row of this table shows the frame number of each video and the total number of frames. The remaining rows show the number of lost frames of each method.

as 'face' if its color $\mathbf{c}(\mathbf{x})$ is closer to the facial color than the background's color:

$$label(\mathbf{x}_i) = \begin{cases} 255(Face) & p_f(\mathbf{c}(\mathbf{x})) \geq p_b(\mathbf{c}(\mathbf{x})) \\ 0(Background) & p_f(\mathbf{c}(\mathbf{x})) < p_b(\mathbf{c}(\mathbf{x})), \end{cases} \tag{9}$$

Then we perform hole-filling to create the face mask as shown in Figure 8 (b). Finally, the face mask is transformed to a cost map using the 3-4 sequential Distance Transform (DT) algorithm [4].

Facial Color Model Update. At frame t, if the shape $\mathbf{s}_t$ obtained by AAM fitting is correct (i.e. the appearance reconstruction error is small enough), but the segmented face region differs significantly from $\mathbf{s}_t$, then the facial color model does not fit to current frame very well. In such case, we update the facial color model based on current AAM fitting. To be efficient, we do not update the background model during tracking.

6. Experiments

We verify the effectiveness of our tracking method by testing the algorithm on ten videos of people downloaded from the Internet. The videos were all taken under uncontrolled conditions. Figure 9 shows the key frames of the videos. The difficulties of tracking these videos stem from illumination changes, large expression and pose changes, fast motions, and cluttered backgrounds.

In our experiments, we trained a multi-view AAM model with multiple bands as explained in section 3.2. For each view's AAM, the shape model contained 8 shape bases, the appearance model contained 40 appearance bases, and the size of the appearance patch was 52×58 pixels.

We combined the trained AAM with different shape initialization methods and constraints and compared the stability of the corresponding six methods: multi-pyramid based AAM fitting (basic AAM) using the previous frame's result as the initial shape without any constraints; AAM fitting using our robust shape initialization method (AAM+RI); three methods that use our shape initialization but different temporal constraints (Liao's work [11] that constrains the matched points' positions (AAM+RI+TP), Liu's work [12] that matches the face appearances (AAM+RI+TA), and our constraint (AAM+RI+TO) formulated as Equation (4)); and finally our temporal matching and face segmentation constrained AAM (AAM+RI+TO+FS). In our experiments, the video resolution was 320×240. The basic multi-pyramid AAM used two levels of pyramid fitting. For our algorithm (AAM+RI+TO+FS), the weights of temporal matching and face segmentation constraints were set to 100 and 0.01, respectively.

We evaluated the stability of a tracking method by counting how many frames were lost by that method. We judged a frame lost if the appearance reconstruction error of the current shape was larger than a threshold. To determine the threshold, we calculated the reconstruction errors of a set of aligned and misaligned shapes (about 2000 samples for each), and set the threshold as the error value that maximally separated the aligned and misaligned samples. In our experiments, the threshold was set to 18.5.

Table 1 shows the statistical data of the six methods on the test video set. As may be seen, for these videos in real life, the basic AAM is not very stable. Our robust shape initialization (AAM+RI) improves the tracker's performance significantly, with the number of total lost frames reduced to 527 from 2098. By adding temporal constraints, three methods ((AAM+RI+TP), (AAM+RI+TA) and our method (AAM+RI+TO)) all improve the tracker's stability, though our method (AAM+RI+TO) lost total 267 frames less than the other two methods. The best results are obtained by our

method (AAM+RI+TO+FS), which combines the temporal matching and face segmentation constraints. By using the combined constraints, only 218 frames are lost out of 50250 total.

Figure 10 shows the typical tracking results of four videos (v1, v2, v6, and v9 in Table 1). Our tracking algorithm accurately localizes the facial components, such as eyes, brows, noses and mouthes, under illumination changes as well as large expression and pose variations.

Our tracking algorithm runs in realtime. On a Pentium-4 3.0G computer, the algorithm's speed is about 50 fps for the video with 320×240 resolution.

7. Conclusions

In this paper, we propose effective approaches to combine temporal matching and face segmentation constraints into AAM-based face tracking. The robustness of the AAM tracker is greatly improved. Our tracker combines the capabilities of a generic AAM model and feature-based temporal matching, and has improved generalizability. By adding face segmentation as a soft constraint, the face outline is more stably tracked. Our tracker shows good performance for real videos.

Currently our tracker cannot robustly track profile views with large angles. We plan to add more AAM models for such views to improve performance. The tracker's ability to handle large occlusion also needs to be improved.

References

[1] J. Ahlberg. An active model for facial feature tracking. *JASP*, 2002(6):566–571, June 2002.

[2] S. Baker, R. Gross, and I. Matthews. Lucas-kanade 20 years on: A unifying framework: Part 4, 2004. Technical Report CMU-RI-TR-04-14, Robotics Institute, Carnegie Mellon University.

[3] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *SIGGRAPH*, pages 187–194, 1999.

[4] G. Borgefors. Hierarchical chamfer matching: a parametric edge matching algorithm. *PAMI*, 10(6):849–865, 1988.

[5] K. Choi, J.-H. Ahn, and H. Byun. Face alignment using segmentation and a combined aam in a ptz camera. In *ICPR (3)*, pages 1191–1194, 2006.

[6] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *ECCV*, 2:484–498, 1998.

[7] T. F. Cootes and C. J. Taylor. Constrained active appearance models. In *ICCV*, pages 748–754, 2001.

[8] T. F. Cootes, G. V. Wheeler, K. N. Walker, and C. J. Taylor. View-based active appearance models. *Image Vision Comput.*, 20(9-10):657–664, 2002.

[9] D. Cristinacce and T. Cootes. Feature detection and tracking with constrained local models. In *BMVC, Edinburgh, UK*, pages 929–938, 2006.

[10] R. Duda, P. Hart, and D. Stork. Pattern classification (2nd edition), 2000. Wiley Press.

[11] W.-K. Liao, D. Fidaleo, and G. G. Medioni. Integrating multiple visual cues for robust real-time 3d face tracking. In *AMFG*, pages 109–123, 2007.

[12] X. Liu, F. Wheeler, and P. Tu. Improved face model fitting on video sequences. In *BMVC07*, 2007.

[13] C.-P. Lu, G. D. Hager, and E. Mjolsness. Fast and globally convergent pose estimation from video images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(6):610–622, 2000.

[14] I. Matthews and S. Baker. Active appearance models revisited. *IJCV*, 60(2):135–164, 2004.

[15] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, USA, August 1999.

[16] J. Pilet, V. Lepetit, and P. Fua. Real-time non-rigid surface detection. CVPR05.

[17] K. Schwerdt and J. L. Crowley. Robust face tracking using color. In *FG*, pages 90–95, 2000.

[18] J. Shi and C. Tomasi. Good features to track. In *CVPR*, Seattle, June 1994.

[19] L. Vacchetti, V. Lepetit, and P. Fua. Stable real-time 3d tracking using online and offline information. *PAMI*, 26(10):1385–1391, 2004.

[20] Q. Wang, W. Zhang, X. Tang, and H.-Y. Shum. Real-time bayesian 3-d pose tracking. *IEEE Trans. Circuits Syst. Video Techn.*, 16(12):1533–1541, 2006.

[21] R. Xiao, L. Zhu, and H. Zhang. Boosting chain learning for object detection. In *ICCV*, pages 709–715, 2003.

[22] W. Zhang, Q. Wang, and X. Tang. Real time feature based 3-d deformable face tracking. In *ECCV (2)*, pages 720–732, 2008.

[23] M. Zhou, Y. Wang, X. Feng, and X. Wang. A robust texture preprocessing for aam, 2008. Proc. of Conference on Computer Science and Software Engineering.

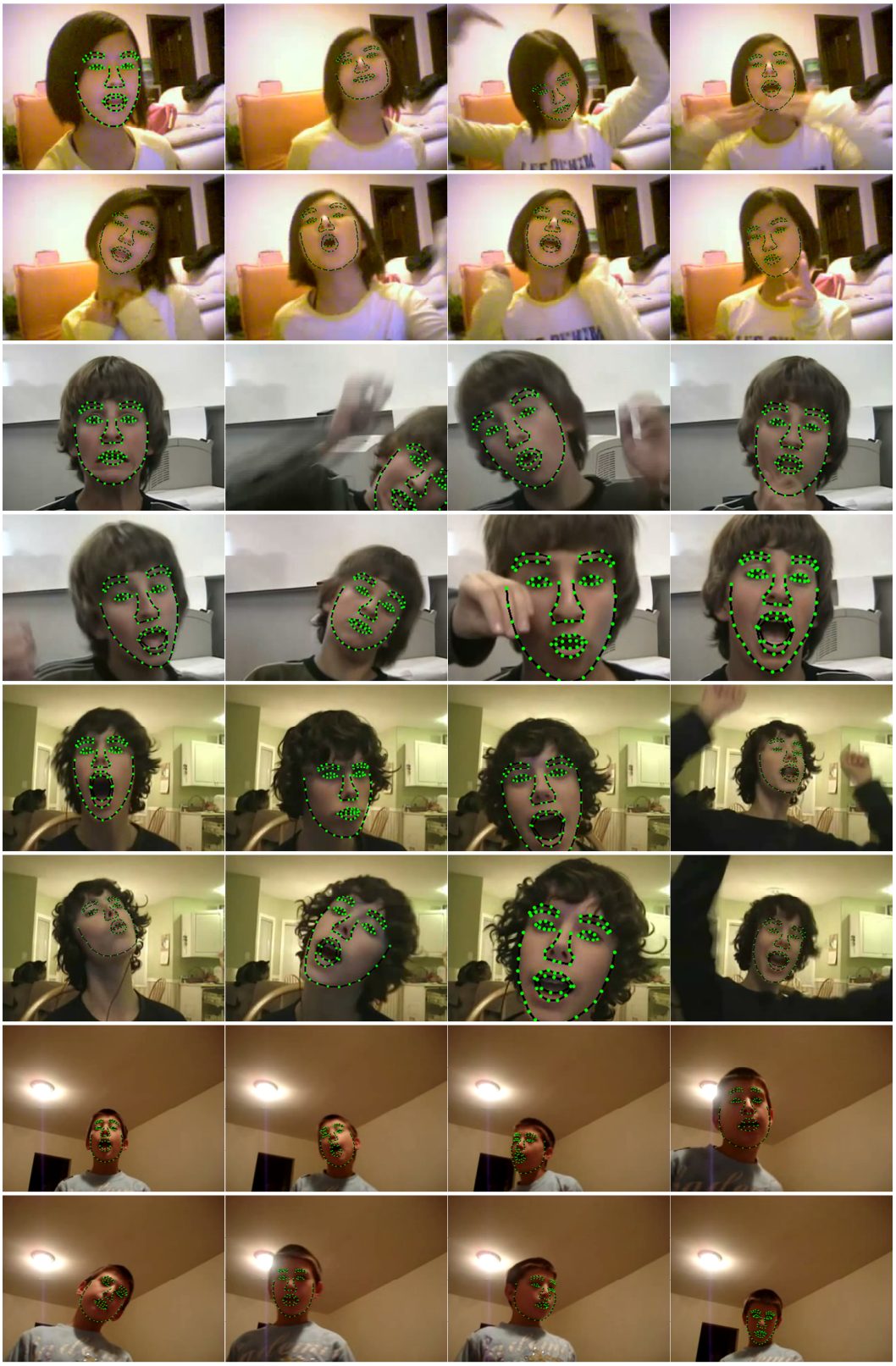


Figure 10. Tracking results of the videos v1, v2, v6, v9 in Table 1.