Face Orientation Estimation in Smart Camera Networks

Chung-Ching Chang

1. Introduction

An important motivation for face orientation analysis derives from the fact that most face recognition algorithms require face images with approximately frontal view to operate efficiently, such as principle component analysis (PCA) [6], linear discriminant analysis (LDA) [4], and hidden markov model (HMM) techniques [2].

In a networked camera setting, the desire for a fontal view to pursue an effective face analysis is relaxed due to the distributed nature of the camera views. Instead of acquiring frontal face image from any single camera, we propose an approach to face reconstruction in a smart camera network by collaboratively collecting and sharing face information spatially.

The framework of spatiotemporal feature fusion for face reconstruction and analysis is shown in Fig. 1. In-node feature extraction in each camera node consists of low-level vision methods to detect features for estimation of face orientation or the angular velocity. These include the hair-face ratio and optical flow, which are obtained through Discrete Fourier Transform (DFT) and Least Squares (LS), respectively. Another feature extracted locally is a set of head strips, which is used to estimate relative angular difference to the face between cameras by a proposed matching technique. A Markov model is designed to exploit the geometic connectivity between strips, and a Viterbi-like algorithm is applied to select the most probable displacement between head strips of the two cameras. Therefore, the proposed technique does not require camera location to be known in prior, and hence is applicable to vision networks deployed casually without localization.

A spatiotemporal feature fusion is implemented via key-frame detection, and a spatiotemporal reinforment learning. The key frames are obtained when a camera node detects a frontal face view through a hair-face analysis scheme and this event is broadcasted to other camera nodes so that the fusion schemes for face analysis can be adaptively adjusted according the the relative angular estimates, in order to maintain a high confidence level. The proposed spatiotemporal reinforment learning cooperate the temporal correlation into the state transition matrix and the spatial correlation into a cost function design, and choose the trellis with the minimum cost.

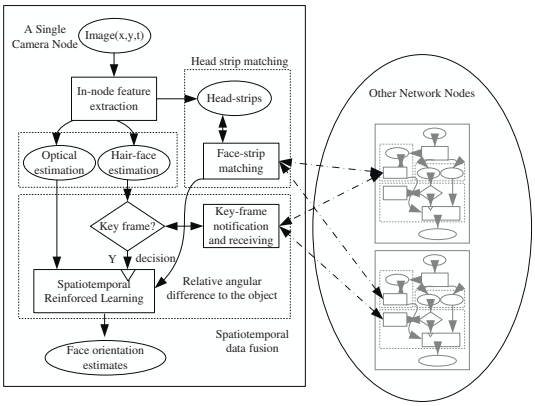

Figure 1. Framework of spatiotemporal feature fusion for face orientation analysis.

2. In-node Feature Extraction

Local data processing algorithms in each camera node consist of low-level vision methods to detect features for estimation of face orientation, including optical flow and hair-face ratio as introduced in the following subsections. These techniques are developed to be of low computational complexity, allowing them to be adopted for in-node processing implementations.

2.1. Optical Flow Estimation

This analysis project the motion of the head into several orthogonal dimensions and estimate the projected vector by least square estimation. In order to reduce computational complexity, we only consider the motion vector of the edges of a face. In our experiments, we assume the head turns without tilt and pan, so we decompose the head motion into only translation and rotation in y axis to simplify the analysis. The decomposition model is as follows:

$$v_i = t + r\omega cos(\theta_i) \tag{1}$$

where v_i is the norm and direction of the motion vector in the direction orthogonal to the head's vertical axis (where positive sign indicates the direction is to the right, and negative sign to the left) , t is the translation factor, r is the transversal radius of the head, ω is the angular motion, and $rcos(\theta_i)$ represents the distance from the point of the motion vector to the longitudinal axis of the head in the 2D

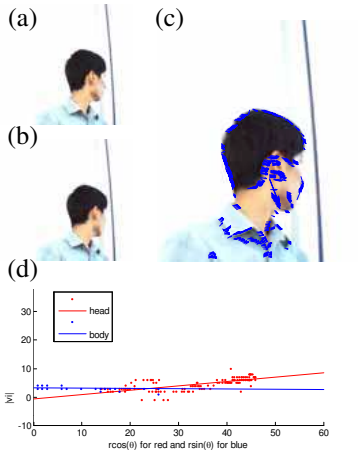

Figure 2. Opical flow estimate (a) image(x,y,t), (b) image(x,y,t+1), (c) image(x,y,t) and the motion vectors, (d) Least squares estimates.

image plane. By placing Eq.(1) in vector form, we have

$$\underbrace{\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}}_{v} = \underbrace{\begin{bmatrix} 1 & rcos(\theta_1) \\ 1 & rcos(\theta_2) \\ \vdots & \vdots \\ 1 & rcos(\theta_n) \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} t \\ \omega \end{bmatrix}}_{z} \tag{2}$$

Minimizing the mean square error of the motion vectors under the model yields the least squares solution of x as:

$$z_{ls} = (A^T A)^{-1} A^T v \tag{3}$$

where the first element of z_{ls} is the translational velocity and the second element is the angular velocity of the head. The residue of the least squares analysis yields a measurement of the confidence of the estimation. Experimental results are shown in Fig. 2, where the slope indicates the angular velocity, and the intersection on y axis indicates the translational velocity.

2.2. Hair-Face Ratio Estimation

To estimate the hair-face ratio, we first classify the head region into face and hair regions by color. There is much previous work on using skin color model and hair color model for face detection [1] [3] [5]. In this paper, we simply apply similar method with value applied well in the sequence. Further research on exploiting model bias between cameras and skin color model estimation based on the EM application (paper evaluation problem) proposed by John Platt.

Based on the hair-face classification, face orientation is analyzed in the following procedures as shown in Fig. 3. Consider the head as a ball in 3D space, and cut the surface

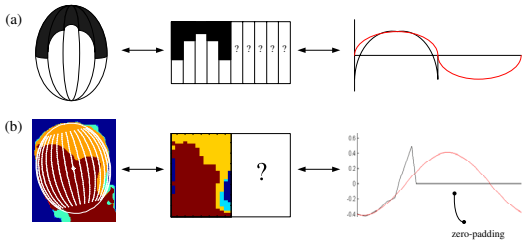

Figure 3. Procedure for the hair-face ratio estimation (Illustration of how the head ellipsoid (right) is transformed into a sequence of hair-face categorized image slices (middle), and into a ratio sequence with zero-padding (left)).

of the ball into N equally spaced strips along its longest axis direction, as shown in figure 3. In each camera frame, we can only see m of the N strips of the ellipsoid. Calculating the ratio of the hair region to the face region in each of the m strips and padding zero to the strips that cannot be seen in the current frame, we form a ratio sequence of length N. By analyzing DFT of the ratio sequence and considering only the phase of the fundamental frequency in the frequency domain, we can estimate the face orientation based on the assumption that the hair-face ratio is symmetric in the frontal face and is approximately a sinusoidal curve along the surface of the ellipsoid.

3. Head Strip Matching and the Relative Angular Difference to the Face between Cameras

Instead of calculating the ratio of the hair region to the face region in each of the m strips, we sample each of the m strips with n samples. Prior to the sampling, a 2D median filter is applied to reduce noise as well as introduce correlation between sampling points and the nearby pixels. Geometrically, if all cameras are deployed at the same horizon, the relative angular difference to the head between two cameras would cause a shift in their strips. Therefore, matching the head strips of the two cameras and finding the displacement of the strips give us the (quantized) relative angular difference to the object between the two cameras at a given time.

The head strip mapping is based on a Markov model and a Viterbi-like algorithm as shown in Fig. 4. Considering two sets of head strips Y and Y', corresponding to the head images captured in two cameras, C and C', let $Y = [y_1 y_2 \dots y_m]$ and $Y' = [y'_1 y'_2 \dots y'_m]$, where $y_i, y'_i \in \mathcal{R}^n$ correspond to n sample points in a single strip. Our problem now is to map the strips in Y' to the strips in Y with the constraint that y_i, y'_i are in some spatial order.

We now introduce the concept of the states S. Let $S = [s_1 s_2 \dots s_N]$ denote all states for the strips of a head (360^o), for example, s_1 representing the strip that includes the nose

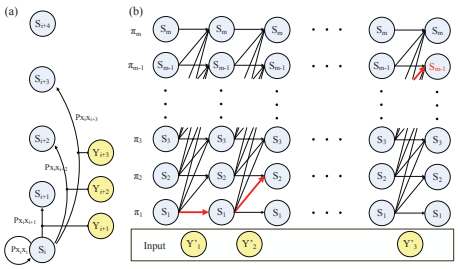

Figure 4. Illustration of the Markov model and Viterbi-like algorithm. (a) The Viterbi-like model generated by the head strip set in camera C, (b) The trellis of the Viterbi-like algorithm. S_{m-1} in the rightmost row is the state with the minimum cost, and the corresponding trellis is marked in thick (red) line.

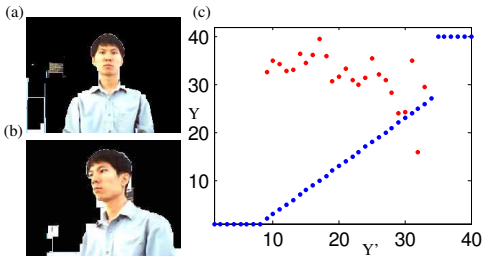

Figure 5. Example of strip matching. (a) Background-subtracted image in camera Y', (b) Background-subtracted image in Y, (c) The trellis with minimum cost (blue) and the corresponding cost in each step of the Viterbi-like algorithm (red).

trail. For each of the captured head images, the corresponding head strips Y should map to a consecutive subset of S, denoted S_Y, which is not known in prior and is approximately of length m. In other words, Y is a representation of the states S_Y. As we scan vertical sampling lines through the head horizontally, we are actually going from state to state, for example , from s_i to s_{i+1}. Ideally we will get y_j and y_{j+1} for certain i and j. However, due to the fact that the head is not a perfect ball, we may as well get y_j and y_{j+k} for certain j and small $k \geq 0$, the latter constraint showing that the two states should be near and cannot occur in a reverse order as we scan through the head. In other words, the probability of Ps_is_{i+1} , the probability of going from the current state to the next state as we scan through the head, is not necessarily 1. The probability of the transition between states forms a Markov model, as shown in Fig.4(a). In our experiment, the choice for the probability is

$$P_{s_i s_{i+k}} = exp(-\frac{(k-1)^2}{2\sigma^2})(u(k) - u(k-4)) \quad (4)$$

where u is the unit step function and σ is the bandwidth parameter. Further normalization is required.

As we match the set of strips Y' to Y, we first assume that the representation Y is ideal, corresponding to the states S_Y one-by-one. Under this assumption, we transform the Viterbi algorithm, a supervised learning algorithm, into an unsupervised way of learning, which we call a Viterbi-like algorithm. For each given input y'_i, we can sum the cost in each of the previous states and the cost-to-go(w) in each branch, and choose the branch with the minimum cost as the path from the previous states to the current states. The cost of the branch w is calculated as follows:

$$w_{s_i s_{i+k}} = -\ln(P_{s_i s_{i+k}} \gamma(y'_{i+k}; s_i s_{i+k})) \quad (5)$$

where $\gamma(y'_{i+k}; s_i s_{i+k})$ is calculated by the inverse of the mean square error between strips y'_{i+k} and y_{i+k}.

The initial states are assumed to be equally likely, meaning that the matching can start from any of the states in S_Y. The first and the last states in S_Y may be regarded physically as the not-in-Y (not in current face) states. Therefore, some exceptions for the probability model are made in the first and the last states, where Ps_1s_1 is given higher probability and Ps_is_m is 1 when $i = m$, and zero otherwise.

According to the Viterbi algorithm, the path with the smallest cost is chosen. For example, as in Fig. 4, assume s_{m-1} in the rightmost column is the state with the minimum cost, and the corresponding previous paths are marked with thick (red) lines, showing that the paths are $[s_1 s_1 s_2 \ldots s_{m-1}]$. In Fig5, an example of head strip matching is shown, the tellis of the Viterbi-like algorithm is shown in the right figure with blue dots, where red dots represent minimum branch cost (w) in each Viterbi-like step. Notice that the trellis, excluding those in states s_1 and s_m, intersect the x-axis around 10, which means the displacement between two head images is 10 strips, or 45 degrees in the example.

4. Spatiotemporal Data Exchange

Collaboration between cameras is achieved by data exchange. Correlations in temporal domain is exploited since face orientation and angular velocity, one being the derivative of the other, in consecutive frames are continuous provided that the time lapse between frames is short. Data exchange in spatial domain would also be helpful in sharing, comparing, and validating data since for any time instance the captured image in each camera should reflect the same motion in 3D.

4.1. Key Frame Detection

Key frames are the frames that include feature or estimates with high confidence. The hair-face ratio based on the phase of the fundamental frequency is sensitive to the face angle. Although the true face angle is not a linear function of the estimates, a frontal view with the hair-face ratio approximately symmtric to the face center can be detected accurately. Under the assumption that head motions

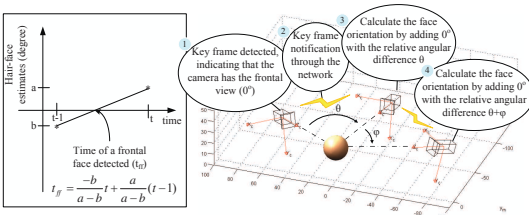

Figure 6. Illustration of the key frame detection procedure

are piecewise linear between samples, the time of a frontal view, defined as a key frame event, can be determined by interpolation. Once a key frame is detected, the time of its detection is notified to other cameras. Since the key frame is associated with relatively high confidence, other cameras would assume the received key frame orientation estimation to be true and calculate the face orientation by adding that with the relative angular difference to the object between themselves and the camera that broadcasted the key frame.

4.2. Spatiotemporal Reinforcment Learning

In our framework, the face orientation between key frames are determined by the accumulation of the angular motions. Recall that the optical flow estimation is a prediction over tens to hundreds of motion vectors According to the law of large numbers, the pdf of the estimation itself will be Gaussian distributed, no matter what the original error distribution is. Assuming that the optical flow estimates in different time are independent, the variance of the accumulated optical flow estimates equals the sum of the variance of each individual optical flow estimates. Therefore, while the orientation estimates are deterministic when a key frame occurs, the orientation estimates between key frames are stochastic with uncertainty increase over time. Hence, we apply a spatiotemporal reinforcement learning algorithm to choose the best trellis with the minimum cost as the uncertainty increases.

In a Markov decision process(MDP), we are usually given the states(S), the state transition probability(P_{sa}), the discount factor(γ), and the cost function(C), and want to determine the optimal policy(π) and its correponding actions(A). In our face orientation estimation settings, S are the (discretized) face orientations, each column of P_{sa} is the probability density function of a Gaussian random variable, with mean and variance determined by the optical flow estimation. The cost function is the sum of the absolute difference of the estimates between cameras. Choosing the cost value to be a L_1 norm of the difference of the estimates can avoid the effect of an outlier. Different from the ordinary MDP, the P_{sa} here may change over time, and thus the action also changes over time. The spatiotemporal reinforcement learning proposed here cooperate the temporal information into P_{sa}, and spatio information into C. Intuitively, the result is similar to that interpolated by the optical flow estimates, however, the corresponbdence between cameras are achieved through the L_1 norm penalization.

5. Comparative Experiments

The setting of our experiment is as follows: Three cameras are placed approximately on the same horizon. One camera (camera 3) is placed in frontal direction to the seat, and the other two are with about $+42^o$(camera 2) and -37^o(camera 1) deviations from the frontal direction. The experiment is conducted with a person sitting still on a chair with the head turning from right(-50^o) to left($+80^o$) and then to the front($+40^o$) without much translational movement. The time lapse between consecutive frames in each camera is half a second, the resolution of the cameras is 320x240 pixels[2].

The result of the estimated relative angular difference to the face between cameras is shown in Fig. 7. The mean and the STD of the estimates in camera 1 are -37.25^o and 17.19^o , and those of camera 2 are $+41.25^o$ and 6.19^o. Both of the time-averaged estimates are close to the ground truth. Four examples of the reconstructed face based on these estimates are shown in Fig. 8. The example in (c) fails to match well in the nose trail region due to an under-estimation of the relative angular differnce to the face between the cameras. In the example in (d), one should notice that the left and the right ears are not in the same horizon, indicating that the in-node signal processing fails to capture the head geometry in fitting an ellipse to the head region, which in turn deteriorates face matching performance.

The results of the collaborative face orientation estimation are shown in Fig. 10 and Fig. 11, with camera locations assumed known and unknown, respectively. The dotted lines in the figures show the ground truth face orientation at each time instance. Although the estimation is degraded without camera location known in prior, the estimates without prior knowledge of the location still predict the face orientation in an acceptable level.

6. Conclusions

In this project, the face model reconstruction and analysis problem is approached in a networked camera setting. Based on the distributed nature of the network, we propose a spatiotemporal feature fusion framework to address the problem which involves two aspects. First, a collaborative technique for head strip matching based on a Markov model and a Viterbi-like algorithm predicts the relative angular differences to the face between cameras and reconstructs a face model without having to know camera locations in prior. Second, spatiotemporal collaboration is embodied through identification and exchanging of key frames, and the design of the state transition matrix and cost function in a re-

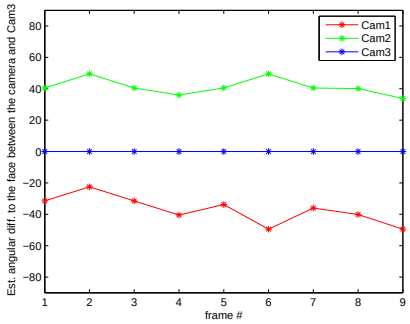

Figure 7. Estimated relative angular differences to the face between the cameras.

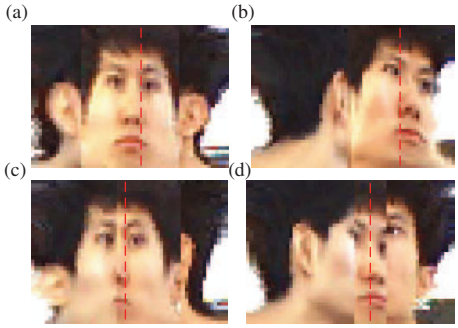

Figure 8. Examples of the reconstructed head model. (a) and (b) Successful examples, (c) Fair example, (d) Unsuccessful example.

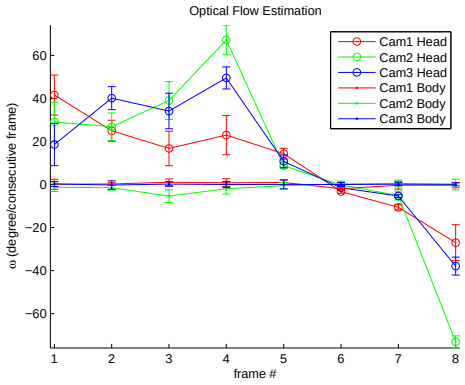

Figure 9. Face angular motion estimation by optical flow estimates.

inforcement learning algorithm. Comparative experiments with and without spatiotemporal collaboration indicate that the proposed technique can successfully predict the face orientation within an acceptable level without prior camera location information.

7. Acknowledgement

I would like to thank Prof. Hamid Aghajan for his help with this project. I consulted him for the design of the framework on the data fusion between cameras. I would also like to thank Chen Wu for her contribution on the real-time optical flow algorithm and her idea on the framework. Furthermore, I would like to thank Prof. Andrew Ng and Daniel Chavez whom I consulted for project direction and machine learning algorithms.

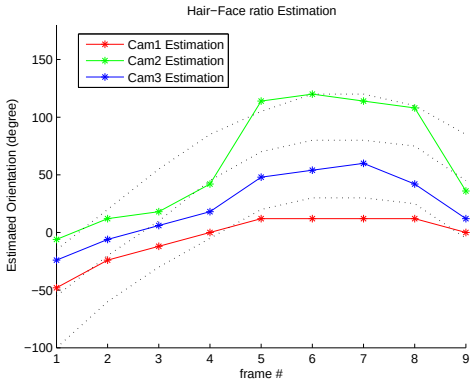

Figure 10. Face orientation estimation by hair-face ratio estimates.

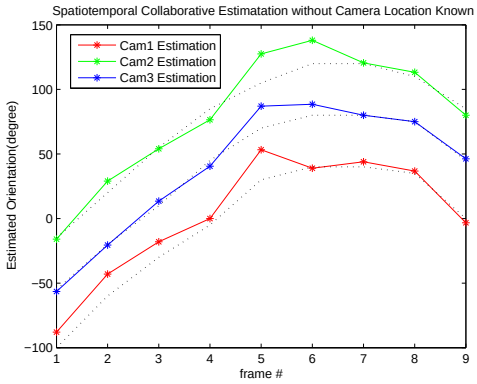

Figure 11. Spatiotemporal Face orientation estimation without knowing camera locations.

References

[1] Q. Chen, H. Wu, T. Fukumoto, and M. Yachida. 3d head pose estimation without feature tracking. In *IEEE Conference on FGR*, 1998. 2

[2] D. Kurata, Y. Nankaku, K. Tokuda, T. Kitamura, and Z. Ghahramani. Face recognition based on separable lattice hmms. In *Proc. of ICASSP*, 2006. 1

[3] B. Kwolek. Face tracking system based on color, stereovision and elliptical shape features. In *IEEE Conference on AVSS*, 2003. 2

[4] C. Liu and H. Wechsler. Enhanced fisher linear discriminant models for face recognition. In *Proc. of ICPR*, volume 2, pages 1368–1372, 1998. 1

[5] F. Liu, Q. Liu, and H. Lu. Robust color-based tracking. In *Proc. of Third Conf. on Image and Graphics*, 2004. 2

[6] M. Turk and A. Portland. Eigenfaces for recognition. *J. Cognition Nueralscience*, 3(1):71–86, 1991. 1