Head Pose Estimation using Motion Subspace Matching on GPU

Nawarat Auttanugune Thanarat H. Chalidabhongse Supavadee Aramvith
Chulalongkorn University, Bangkok, Thailand
E-mail: thanarat.c@chula.ac.th Tel: +66-2-218-6955

***Abstract*— The head pose estimation is a process of recovering 3D head position in term of yaw, pitch and roll from 2D images. However, the reduction of information from 3D to 2D leads to an ill-posed problem. In this paper, we propose a novel algorithm of head pose estimation that includes facial features tracking for Thai sign language recognition. In order to estimate head pose correctly, feature points tracking requires high precision. Nevertheless it is difficult for low cost cameras where input image quality may be generally poor. To overcome this problem, we introduce an automatic camera signal calibration such that the features can be tracked correctly despite the quality of the input image sequences. Finally, as our approach bases on the state space searching, the local minima problem is common. Hence, we divide the search space into sub spaces and perform parallel computation on GPU.**

I. INTRODUCTION

The natural languages play an important role in human communication. Their structures, grammars, and vocabularies vary from country to country. Meanwhile, sign language or gesture that express through body movement and facial expression is fairly common. For example, we usually turn our head to the direction that we are interested in or talking about, or raising eyebrow when having doubt. Hence, making a computer able to understand these actions is beneficial for interaction between human and computer due to its independency of natural language.

The head pose estimation is a process of recovering 3D head position in term of yaw, pitch, and roll from 2D images. The challenge is that the information of human head in 3D is reduced to 2D when captured with 2D imaging devices. This information reduction leads to *ill-posed* problem. Furthermore, facial features are deformable. Their shapes vary by feeling being expressed so it is difficult to instruct computer to remember specific shapes or characteristic of facial features.

Our previous work [1] presents a simple but effective algorithm of head pose estimation. The main concept can be explained as follows. Firstly, optical flows between 2D input images and projected images generated from rotation of 3D facial model, CANDIDE [2] are matched. The direction of CANDIDE model rotation is based on the error function. The matching of optical flows computed from these two sources determines rotation parameter that is considered as an answer state. The motions being compared between these two sources based on gross head calculation. However, from our experiment, we found that the gross head motion is good when head is rigid. However the algorithm fails when facial features are deformable because they also generate additional flows. Hence, in order to do facial expression extraction, we amend image optical flow calculation from the gross head to the specific facial feature points. Details will be discussed in the following section.

In addition to the first issue, errors from feature point detection can cause large errors of head pose estimation in each frame. There are many causes of inaccuracy such as noise from input images, and unstable camera signal. In this paper, we also propose an automatic camera signal calibration to stabilize point tracking and improve accuracy of head pose estimation.

The paper is organized as follows. Section 2 discusses low level facial feature detection. Section 3 discusses the automatic camera signal calibration. Section 4 discusses head pose estimation with parallel subspace computation on GPU and facial expression extraction. Finally, Section 5 concludes the experiment results and discusses future works.

II. FACIAL FEATURES DETECTION

A. Face Detection and Tracking

We reduce amount of calculation by finding region of the face and tracking this region of interest, ROI, along the input image sequences. The process begins with Haar-like features [3] scanning on the frontal face input image to roughly estimate the face ROI. Let the region of the detected face in an input image denoted by $R(x,y)$ with $R_{00}(x,y)$ and $R_{11}(x,y)$ as its top-left and lower-right corners, respectively. Once we can roughly estimate $R(x,y)$,we perform tracking using the Camshift algorithm [4]. All remaining tasks of the algorithm process inside area of $R(x,y)$. The Fig. 1 illustrates result of the face detection and tracking.

Fig. 1. Face detection and tracking

B. *Facial Feature Detection*

Our previous work utilizes facial feature points only in the initialization process. They are used to adjust size of the CANDIDE model. However, in order to extract facial expression, the facial feature locations are required. Jingying Chen and Bernard Tiddeman [5] proposed an algorithm for locating facial features using skin color thresholding. Their algorithm works automatically in real time. However, the six facial feature points of pupils, nostrils and lip corners are not adequate for facial expression extraction. Dimitris Metaxas, Atul Kanaujia and Peng Yang [6] proposed an algorithm of real time expression recognition. Their algorithm bases on active shapes for localizing facial features on the face. Even though this approach achieves high accuracy of facial expression extraction, but the result of head pose and facial expression cannot be isolated into the values of yaw, pitch, and roll.

In this paper, we perform feature point tracking approach as mentioned earlier. There are 19 feature points. These facial feature points can be divided into 2 groups, global features and local features. The difference between these two groups is that the global features are used to estimate the gross head movement while the local features are used in the facial expression extraction.

The global feature points compose of four horizontal eye corners and a nose tip. These control points are used in gross head movement estimation because they are fairly rigid. Their locations are not varied by facial expression. On the other hand, the remaining 14 points, which compose of six points from eyebrows, four points from vertical eyes corner and four points from the mouth, are facial features that their locations can be varied by facial expression. As illustrated in the Fig. 2, the global feature points are marked in red and those remaining local feature points are marked in green.

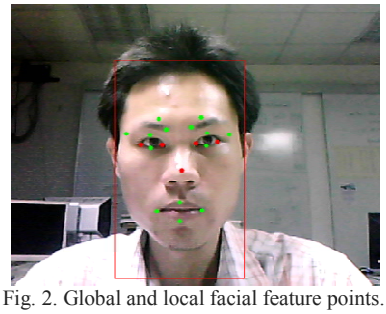

Fig. 2. Global and local facial feature points.

All feature points are initially located by Active Shape Model (ASM) developed by Tim Cootes and Chris Taylor [7]. However, instead of implementing ASM directly, we reduced the number of feature points from over 60 points to only 19 points. This reduction makes our proposed method feasible for real-time application while the impact to tracking precision can be kept minimal. These points can fit 3D CANDIDE facial model well. Each feature point corresponds to one vertex of the CANDIDE model as illustrated in Fig. 3.

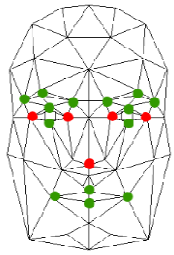

Fig. 3. Correspondences between CANDIDE vertices and facial feature points.

Once all required feature points are detected, we then perform tracking along the input image sequences. Given a location of the detected feature point in an input 2D image denotes as $\kappa_p(x,y)$, the tracking is considered as track lost when $\exists k_p, R_{11}(x,y) < k_p(x,y) < R_{00}(x,y)$.

III. AUTOMATIC CAMERA SIGNAL CALIBRATION

Once we have found all necessary feature points from the previous step, the next step is to track these points along video sequence. In this paper, we utilize the optical flow of Lucas Kanade [8] for point tracking. As discussed in the introduction, an accuracy of feature point tracking is vital because a minor error in feature point detection or tracking can cause large error in the head pose estimation. Also, as our approach bases on tracking method, any error occurred in each consecutive frame will be accumulated through the entire video sequence. From our scenario, we have found that one of the most important factors that cause error in point tracking is signal instability of the imaging device itself.

In general, this problem can be fixed using two phases of point prediction and correction such as using Kalman's filter [9] to enhance accuracy of feature point tracking. Zhiwei Zhu and Qiang Ji [10] proposed a method of facial features refinement through Gabor wavelet coefficient vectors. Their algorithm is based on point prediction and correction. The conceptual idea is that any tracked facial features are refined by matching with the training set in the Gabor space to eliminate any possible error caused by the appearance changes. However, this method requires a large training data set and extra steps of prediction and correction for every input frame. In this paper, we introduce an automatic camera signal calibration which does not require any configuration or prior training data set.

Assuming that the object sits still, then any location changes of the feature points in input images is caused by camera signal instability. Given a feature point p in frame j is denoted as $\kappa_p(j)$, its distance between two consecutive frames is denoted as $\|\kappa_p(j,j+1)\|$. The automatic camera signal calibration is an attempt to tune parameter t of the calibration function which denoted as $\xi(t)$ to yield $\xi(t)\cdot\|k_p(j+1,j+2)\| - \|k_p(j,j+1)\| \approx 0$. Given that the same facial feature point is collected for n times by tracking process,

the formal definition of the calibration function applied for the number of m feature points is defined as in

$$\xi(t) = \sum_{p=1}^{m}\left(\sum_{i=1}^{n-1}\left(\left\|k_p^i(j,j+1)\right\|\right)\left[\frac{2e^{-t}}{1+e^{-t}}\right]\right) \quad (1)$$

Fig. 4 shows the calibration function that attempts to cover most of different feature points. The horizontal axis represents the number of feature points. The vertical axis represents the distance. The red and blue points represent $\left\|k_p(j,j+1)\right\|$ and $\left\|k_p(j+1,j+2)\right\|$, respectively. In this case, $\xi(\beta)$ yields better result than $\xi(\alpha)$. The optimal parameter is $t=\beta$ in this example.

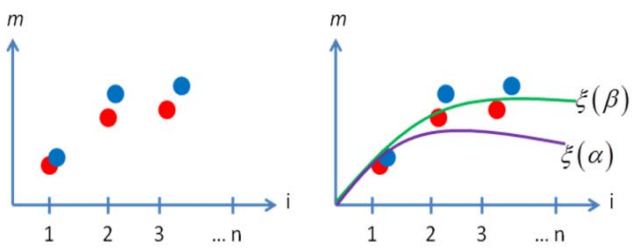

Fig. 4. Illustration of calibration function.

In practical applications, to calibrate the camera signal, the user is required to sit still for a while. At very first input frames, the system attempts to detect and track all feature points under the assumption that any location change of the feature points are caused by camera signal instability. In our case, the system collects all feature points for 1 second at 30 frames/s which equals to 19*30 = 570 points then keeps tuning for the optimal parameter t.

In Fig. 5 (left), the system first displays all detected feature points as white. Once it can estimate the optimal parameter t, all feature points are represented as red and green points as shown in Figure 5 (right). There are some cases where the system cannot find the optimal parameter t. This happens when the object has some movements which causes large motion feature points and the iteration of estimating the optimal parameter t is out of calibration range, $0<t<1$, the system will restart process of feature points detection, tracking, and calibration automatically.

Fig. 5. Automatic camera signal calibration.

IV. HEAD POSE ESTIMATION AND FACIAL EXPRESSION EXTRACTION

A. *Face Detection and Tracking*

With reference to our previous work, the head pose is estimated from the comparison between motion vectors of 2D input image and the 3D facial CANDIDE model's rotation. It is controlled by parameters: yaw $=\gamma$, pitch $=\beta$ and roll $=\alpha$. The parameter values that yield the minimum output of error function is considered as an answer. In this paper, as we reduce motion from gross head to a set of specific global feature points, each point from 2D input image corresponds exactly to a specific CANDIDE model's vertex. All image motions are now multiplied by stabilization factor. We redefine the input image motion's notation from I_i to $\tilde{I}_i$ such that $\tilde{I}_i = I_i \cdot \xi(t)$. Hence, the error function's definition is redefined as

$$E_k(\gamma,\beta,\alpha) = \sum_{i=1}^{n}\left[\min\left(\sqrt{\left\|\tilde{I}\right\|_i^2 + \left\|C\right\|_i^2 - 2\left\|\tilde{I}\right\|_i \left\|C\right\|_i \cos(\theta_i^I - \theta_i^C)}\right)\right] \quad (2)$$

However, as this approach bases on the state space search so it inherits common problems of the local minima. As illustrated in Fig. 6, there are possibilities that global minima may not align with the optimal search path. If this situation occurs, the estimation will be less accurate and the error will get accumulated along video sequences.

There are many techniques that can be applied to solve this problem such as gradient descent and backtracking. But these approaches are not applicable with real time applications due to considerable processing time in some cases. Thus, we propose a solution by dividing the search space into sub spaces and perform computation in parallel mode on GPU as illustrated in Fig. 7.

In Fig. 7, instead of always traverse through the minimum error state, we expand all possible combination of yaw, pitch, and roll at the first level then treat each state as a root node of subspace. For each subspace, we use the same process of state expansion and selection as discussed earlier. At the end of execution, each subspace returns the minimum error state as a local result. Given the optimal local result of subspace L denoted as e_k^L, the global result which denoted as E_k is computed by CPU on all e_k^L that $E_k = \min_{1<L<n}\left(e_k^L\right)$.

B. *Facial Expression Extraction*

Once the gross head pose is estimated, the facial expression extraction is trivial because facial feature points are located on surface of the head. The depth can thus be considered as a constant value along input image sequences. Given the rotated 3D CANDIDE model by $E_k(\gamma,\beta,\alpha)$ at input image frame i denoted as S_i, the motion vector of a local facial feature point at input image frame i denoted as $l_i = [\Delta x \quad \Delta y]^T$, the new location of its corresponding vertex on S_i, denoted as $c_i = (X,Y,Z)$, equals to $\hat{c}_i = (X+\Delta x, Y+\Delta y, Z)$.

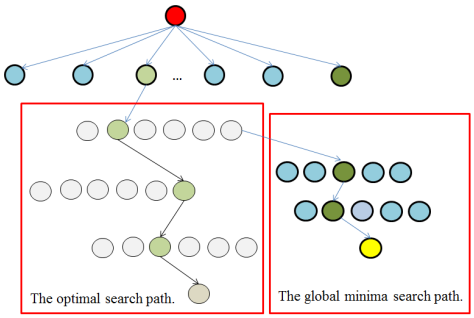

Fig. 6: Misalignment between the global minima and the optimal search path.

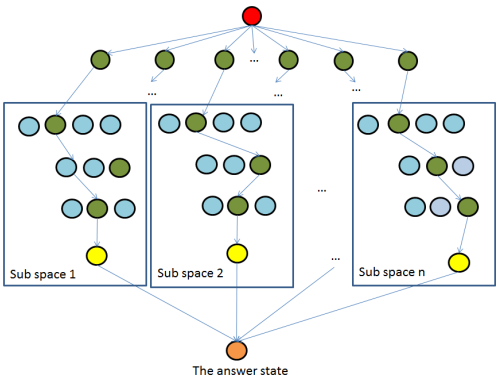

Fig. 7: Parallel subspace searching on GPU.

V. EXPERIMENTAL RESULTS

We observe the result of head pose estimation without calibration by asking the subject to sit still for a moment. Figure 8 reveals that without the camera signal calibration, the estimated value of yaw, pitch, and roll are oscillated up and down in a limited range. Also, as our algorithm bases on tracking approach, any error of yaw, pitch, and roll occurred are getting accumulated along the input image sequences as illustrated in Figure 9.

Then we implement the automatic calibration function and perform the testing. The result shows that tracking points become more stable. Not only this approach stabilizes the signal but also keeps the flexibility of the calculation of the motion. Error! Reference source not found. illustrates the experimental results and the Fig. 12 illustrates the application with 3D graphic animation.

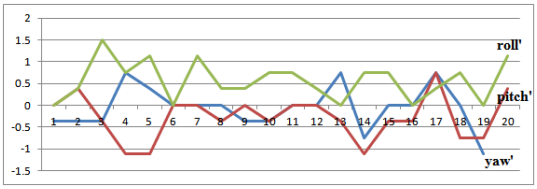

Fig. 8. Errors from head pose estimation from camera signal instability.

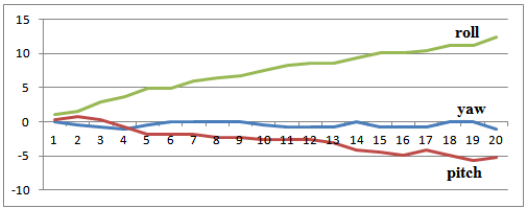

Fig. 9. Error accumulation in tracking.

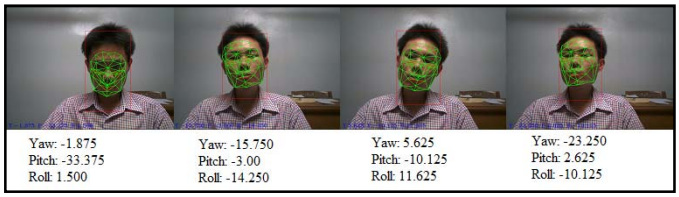

Fig. 10: Experimental result

Error! Reference source not found. shows the comparison time profiling of head pose estimation implementations on CPU and GPU per single input frame. The testing is performed on a laptop with AMD A6-3400M, 8 GB memory and AMD Radeon HD 6520G. We discover that the subspace search implementation using GPU always returns the minimum error state which is mostly similar to CPU's computation as illustrated in Fig. .

However, there are possibilities of more than one minimum error state be found. Under this situation, the GPU will always select the shortest part as an answer state which may not be the same answer state when performing calculation on CPU. As illustrated in the Fig. **11**, the estimated rotation angle from CPU and GPU at the input frame 206 is the same but is different in direction.

VI. CONCLUSIONS

Even though the proposed algorithm can estimate head pose along with facial expression extraction correctly, there are still rooms for the enhancement. One enhancement can be the determination of feature point re-initialization. The current system performs re-initialization only when there is a feature point that leaves out off the face ROI and when feature points are distorted such as the face is occluded by hands. In future works, we would like to define a threshold function that can be used to determine whether the structure of feature points is

still intact or not. Finally, the 3D facial model can be improved. We currently apply the original version of CANDIDE that has just two movable vertexes of mouth. We plan to replace the model by the 3rd CANDIDE version [10] which has larger number of mouth vertexes. This will allow better mouth animation.

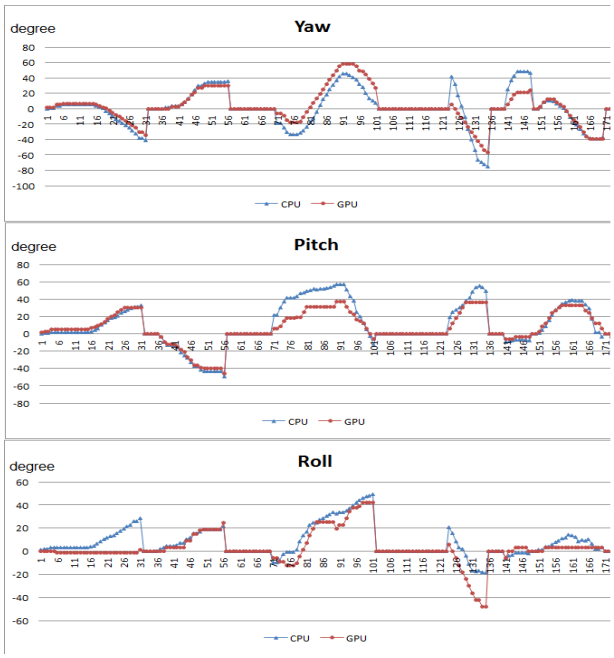

Fig. 11. Result comparison between CPU and GPU

Table 1 Execution time profile

	Elapsed (microseconds)	Number of states	Avg. time per state (microseconds)
CPU	1,317.86	84	15.69
GPU	2,143.14	208	10.30

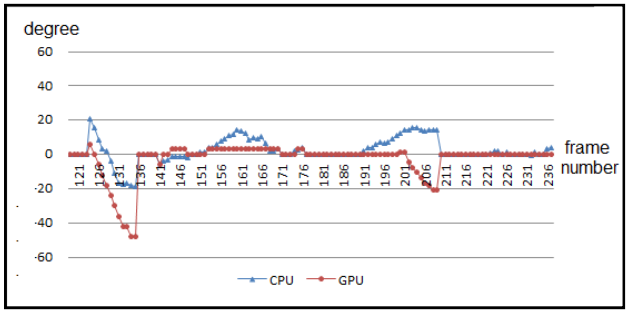

Fig. 11. Different answer between CPU and GPU

ACKNOWLEDGMENT

Special thanks to Prabhas Chongstitvatana for review and suggestion.

REFERENCES

[1] N. Auttanugune and T. H. Chalidabhongse, Head Pose Estimation using Motion Matching. in *15th International Workshop on Advanced Image Technology*, (Ho Chi Minh City, Vietnam, 2012)

[2] M. Rydfalk, CANDIDE, a parameterized face, Report No. LiTH-ISY-I-866, Dept. of Electrical Engineering, Linkoping University, Sweden, 1987.

Fig. 12. Application with 3D graphic animation

[3] Viola and Jones, Rapid object detection using boosted cascade of simple features. in *Computer Vision and Pattern Recognition*, 2001

[4] Bradski, Gary R, Santa Clara, and Intel Corporation. , Computer Vision Face Tracking For Use in a Perceptual User Interface, Interface 2.2, 1998, 12–21.

[5] J. Chen and B. Tiddeman, A Robust Facial Feature Tracking System. in *Advanced Video and Signal Based Surveillance*, (Como, Italy, 2005), 445-449

[6] D. Metaxas, A. Kanaujia, P. Yang, Facial Features Tracking for Gross Head Movement Analysis and Expression Recognition

[7] T.F. Cootes, C.J. Taylor, D.H. Cooper, J. Graham, Active shape models - their training and application. *Computer Vision and Image Understanding*, *Volume* 61, 38–59

[8] B. Lucas, T. Kanade, "An Iterative Image Registraton Technique with an Application to Stereo Vision," Proc, of 7th International Joint Conference on Artificial Intelligence (IJCAI), pp. 674-679, 1981

[9] G. Welch, G. Bishop, An Introduction to the Kalman Filter, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA. 1995

[10] Z. Zhu and Q. Ji, Robust Pose Invariant Facial Feature Detection and Tracking in Real-Time. in *18th International Conference on Pattern Recognition*, (Hong Kong, 2006), 1092–1095

[11] J. Ahlberg, CANDIDE-3 -- an updated parameterized face, Report No. LiTH-ISY-R-2326, Dept. of Electrical Engineering, Linkoping University, Sweden, 2001.