Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 9729014, 8 pages
https://doi.org/10.1155/2018/9729014

Research Article

A Research on Fast Face Feature Points Detection on Smart Mobile Devices

Xiaohe Li, Xingming Zhang, and Haoxiang Wang

College of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China

Correspondence should be addressed to Haoxiang Wang; hxwang@scut.edu.cn

Received 1 September 2017; Revised 12 December 2017; Accepted 28 December 2017; Published 25 February 2018

Academic Editor: Zhaolong Ning

We explore how to leverage the performance of face feature points detection on mobile terminals from 3 aspects. First, we optimize the models used in SDM algorithms via PCA and Spectrum Clustering. Second, we propose an evaluation criterion using Linear Discriminative Analysis to choose the best local feature descriptions which plays a critical role in feature points detection. Third, we take advantage of multicore architecture of mobile terminal and parallelize the optimized SDM algorithm to improve the efficiency further. The experiment observations show that our final accomplished GPC-SDM (improved Supervised Descent Method using spectrum clustering, PCA, and GPU acceleration) suppresses the memory usage, which is beneficial and efficient to meet the real-time requirements.

1. Introduction

Face feature point detection and tracking is a hot topic in the research field of computer vision. It is the foundational technology of face pose evaluation, face expression transfer or synchronization, and so on. With the popularity of smart mobile terminals, a real-time tracking of facial feature points in mobile terminal is highly on demand. For example, the recently designed mobile [1] or vehicle [2] social networks need high quality face feature points tracking and detection for security authentication. It may also benefit the information retrieval within mobile crowd sensing architectures [3, 4]. However, due to the limited computational capacity of floating points on mobile terminal and the battery power, a series of face feature detection algorithms, which were successful on computer, cannot be realized directly on mobile terminals. Thus we come to the problem about how to realize real-time detection and tracking of face feature points on mobile terminals with limited computation power.

Nowadays, numerous research studies of facial feature point detection have been embodied in the experimental exploration on personal computer (PC) side. According to literature [5], from the model design, the method of facial feature point detection methods can be divided into local facial feature based, global facial feature based, and the hybrid method combining local features and global features. In comparison with other methods developed in recent years, four local feature based algorithms, such as Supervised Descent Method (SDM) [6], Coarse to Fine Shape Searching (CFSS) [7], Ensemble Regressing Trees (TREES) [8], and Explicit Shape Regression (ESR) [9], have achieved outstanding performances. Surprisingly, although these methods do not take into account the geometric features of facial feature points, they still manifest a good expression of facial occlusion. In spite of the simplicity, the speed and accuracy of SDM algorithm are still very competitive so far. And the cascade regression method firstly contributed by SDM has been continuously learned by the scholar community [10–13] until now. Besides, based on a local receptive field of artificial neurons, Baltrusaitis et al. [14] in CMU proposed a method of facial feature point detection, which performed well under different illuminations. Based on both global and local feature descriptions, Hasan et al. [15] developed a shape regression method, which made a breakthrough on the W300 test set. Jaiswal et al. [16] proposed Gabor-LBP based face feature detection using model selection and support vector regression. According to the experiment observations, this method plays a superior role in capturing the face feature points with local occlusion and background clutters. Specially, Martinez et al. [17] broke the traditional routines of sequential regression method on this problem and proposed multivisual cues aggregation regression approach. With the continuous success of deep learning in computer vision tasks, Jourabloo and Liu [13] employed CNN as both shape and coefficients of 3D face model estimators for large pose variant face alignment and it can estimate 3D face shapes as shown in their experiments.

Recently, mobile devices have attracted tremendous attentions. Choi et al. [18] developed a face feature detection system for real-time training on mobile phones. Tresadern et al. [19] introduced the fixed-point method [20] and Haar-Like feature into the Active Shape Model [21] (ASM) algorithm to perform real-time tracking of facial feature points. Based on hierarchical model, Jian-kang et al. [22] proposed a detection method, achieving better results on mobile phones. Jiang [23] developed a facial feature detection system based on Android platform. With facial point detection on Android platform, Hp [24] invented a 3D human face pose estimation system.

Despite the numerous studies on the face feature detection algorithm for mobile terminals, it is still in its infancy, and most of them are based on mobile operating system. In this article, we improved SDM algorithm [6] with two aspects: the calculation and model. First of all, we reduced the dimension of features needed for localizing the facial feature points by using spectrum clustering. Second, we decreased the number of linear models in SDM by PCA. Finally, the local feature extraction and SDM model calculation were combined and optimized on the mobile GPU to further improve the performance of SDM algorithm. Since the SDM method is widely learned by most recently released face feature point detection methods, our modification should be also meaningful to boost the performance of other approaches.

2. Solution Pipeline

Figure 1 presents the schematics of collecting video data with mobile phone cameras. Face feature point detection is applied to a frame extracted from the video stream. Face detection is performed using Viola's face detector. In order to overcome the illumination problems, the preprocessing steps have been employed in this work (details are available in Section 3). The local image feature extraction algorithm is performed on the initial feature points distribution, using LBP, HOG [25], SIFT [26], and so on. The feature extraction algorithm is implemented following GPU parallel design, which is copied into shared memory according to the pixel values of face images needed by each computing unit. At the same time, the SDM algorithm model is also embedded into the shared memory according to the computing unit. The calculated results of each computational unit are the regression components of human face shape.

After obtaining the regression components, we merge the regression components and update the shape model based on the integrated regression components. The process is repeated until that the shape model satisfies the error requirements in the training process.

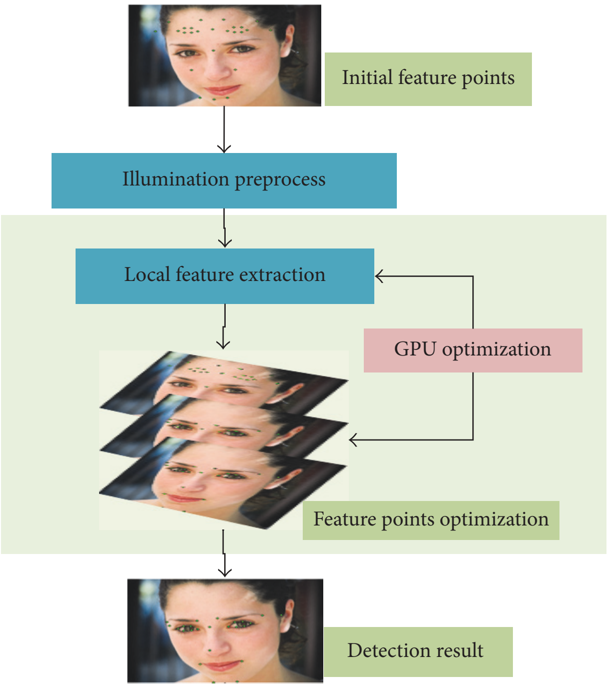


Figure 1: The solution pipeline.

3. Preprocessing

Considering the real video data acquisition, camera may have noise in dim condition, resulting in large image gradient or irregular filter responses, which severely influence the robustness of the detection algorithm. We normalized the face image from two aspects: noise reduction and brightness stretching as shown in Figure 2.

Firstly, we performed median filter to reduce the noise probably caused by the low light conditions. Then the histogram equalization is performed, in order to unify the light effect, clear the surface details of dim face image, and prepare for the subsequent steps.

4. Optimized SDM

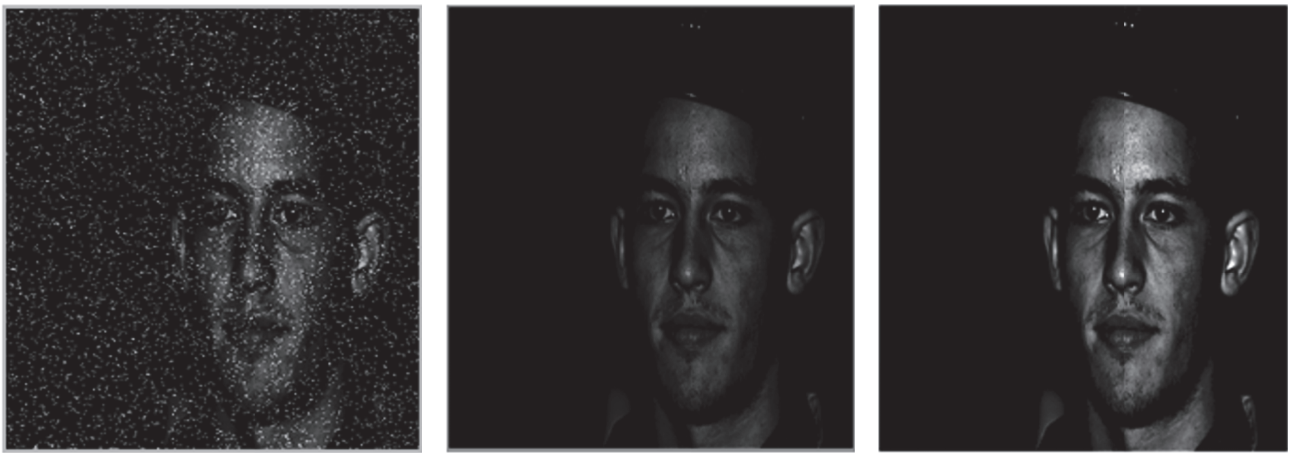
4.1. Introduction to SDM. The core idea of SDM algorithm is to minimize the object function (1), aiming at finding the right modification of current position to the ground truth. The expression is as follows:

$$\Delta x = \arg\min_{\Delta x} \left\| h\left(d\left(x_0 + \Delta x\right) - h\left(d\left(x^*\right)\right)\right) \right\|, \qquad (1)$$

where x^* represents for the labeled facial feature points position; d stands for face images; h is the local description extracted at certain positions; x_0 is the initial shape. According to Newton method, minimizing (1) yields

$$\Delta x = -2H_h^{-1} J_h \left(h\left(d\left(x^*\right)\right) - h\left(d\left(x_0\right)\right)\right), \qquad (2)$$

where H_h is the Hessian Matrix of feature extraction function denoted by h; J_h is the Jacobian of h. Although the function is



Noisy dark image

Median filter

Histogram equalization

Figure 2: Face image preprocessing.

ill posed since the feature extraction function h is commonly unknown in practice, it can be reformulated as

$$\Delta x = R * h\left(d\left(x_0\right)\right) + b. \tag{3}$$

Since the facial feature points are labeled at each instance, $j = 1, 2, \ldots, n$, the difference between initial and optimal feature points Δx^j is also known; thus we have the following linear system:

$$\Delta x^j = R\left(h\left(d^j\left(x_0\right)\right)\right) + b. \tag{4}$$

Let d^j represent the j_{th} training image; Δx_j is used to solve R and b in the training phase. After the $i-1$ regression model R_i, b_i is obtained; we update each initial shape of each individual training samples:

$$x_i^j = x_{i-1}^j - R\left(R_{i-1}\left(h\left(d^j\left(x_{i-1}^j\right)\right)\right) + b_{i-1}\right). \tag{5}$$

Training was accomplished after obtaining the regression models: $\{R_i, b_i\}$, $i \in 1, 2, \ldots, K$.

In the research of facial feature point tracking, SDM algorithm is still applicable. The difference between the face feature points detection and the face feature points tracking is in the initial feature points distribution. In the training step of facial feature points tracking, each individual's initial facial points position x_0^i is randomly sampled once or several times from its label x^{*i} within 95% or 20 pixels rescale or translation. Then, the tracking models are obtained with the same process discussed above. It is reasonable because the face shapes in consequence frames change slowly in the video.

4.2. Model Compression. Through the above analysis, we can find that SDM algorithm uses multiple regression to achieve the face feature point alignment, and its computational complexity is mainly concentrated in the calculation of regression models:

$$\begin{aligned} R_t &= X\left(S_t\right) A_t + B_t, \\ S_{t+1} &= S_t + R_t. \end{aligned} \tag{6}$$

Assuming S_t is the feature points position after t times regression, considering we have 66 feature points to align, we will need $66 * 2 = 132$ linear model at each regression model. However, with careful consideration, we can find that if two feature points y_i, y_j in shape S are not strongly correlated, the local feature extracted at y_i does not help much to locate y_j. Thus, we separate S to $S_1, S_2, \ldots, S_k$, making feature points in the same subshape mostly correlated, and feature points in different subshapes not that relevantly. In order to accurately estimate the number of categories, we introduce spectral clustering for analysis. First, the correlation coefficients between different feature points are defined as

$$\rho_{k,j} = \frac{1}{D}\sum_d \left| \frac{E\left(p_{k,d} - E\left(p_{k,d}\right)\right) E\left(p_{j,d} - E\left(p_{j,d}\right)\right)}{\sigma_{k,d}\sigma_{j,d}} \right|. \tag{7}$$

Let n be the number of training samples, D be the dimension of feature points, and $\sigma_{k,d}$ be the covariance of the k_{th} feature point in dimension d.

We use the correlation coefficient matrix directly to define the Laplacian Matrix:

$$L = D - W. \tag{8}$$

The weight matrix $W_{i,j} = e^{-\rho_{i,j}}$, and $D_{ii} = \sum_j W_{i,j}$. According the spectrum clustering, the number of smallest Eigen values of L minus 1 is the number of catalogs denoted as K. Thus we can get K subshapes by performing k-means on the last second Eigen vector to the $K + 1$ Eigen vectors. Assuming that face feature points can be divided into K classes, the facial feature point location of SDM algorithm can be rewritten as

$$\begin{aligned} S_k^t &= X\left(S_k^{t-1}\right) R_k^T + b_k^T + S_k^{t-1}, \\ S^t &= \bigcup_k S_k^t. \end{aligned} \tag{9}$$

Equation (9) uses union of subshapes' regression model to determine the whole shape of face. The method reduces the length of each linear model $1/K$ times to the original and relaxes the computational complexity of SDM. However, the number of linear models in the framework of the SDM algorithm has not been reduced. Therefore, we use PCA to further simplify the SDM cascade linear model. The process is as follows:

Assuming we have the following linear model:

$$XA + B = Y, \quad Y = (y_1, y_2, \ldots, y_m) \tag{10}$$

we perform dimension reduction using PCA on Y so that $YW = \widetilde{Y}$. $W \in R^{m\times n}$ is the principals of Y and $n \ll m$. We have

$$\widetilde{Y}W^T = Y\left(WW^T\right). \tag{11}$$

If we denote $W^{\text{full}} = [W, W^r]$, this yields

$$\begin{aligned} Y - YWW^T &= YW^{\text{full}}W^{\text{full}T} - YWW^T \\ &= Y\left(WW^T + W^rW^{rT}\right) - YWW^T \\ &= YW^rW^{rT}. \end{aligned} \tag{12}$$

From matrix decomposition of Y^TY, we have

$$W^rW^{rT}Y^TYW^rW^{rT} = W^r\Lambda^RW^{rT}. \tag{13}$$

Λ^R is the diagonal matrix with Eigen values corresponding to W^r. Because $\Lambda^R \approx \mathbf{0}$, we can know $Y - YWW^T \approx \mathbf{0}$. So (11) can be rewritten as

$$XA + B = \widetilde{Y}W. \tag{14}$$

Since W is orthogonal, multiplying W^T on both sides of (14), we get

$$XAW^T + BW^T = \widetilde{Y}. \tag{15}$$

It can be rewritten as

$$X\widetilde{A} + \widetilde{B} = \widetilde{Y}. \tag{16}$$

From the analysis above, we can know that the width of each linear model can be reduced to number of principal of Y.

4.3. GPU Acceleration. In the design principle of SDM algorithm, we know that the robustness of local feature extraction from facial feature points has a very important impact on the effect of face alignment. In practice, we find that the feature extraction and the model computation consume half of the time cost of SDM algorithm, respectively. With the development of computer vision, scholars have proposed an endless stream of local feature descriptors. Among them, the most widely used features are HOG (Histogram of Gradients) [25], SIFT (Scale Invariant Feature Transformation) [26], and LBP (Local Binary Patterns) [27]. However, these features are either too time consuming or difficult to be parallelized. For example, HOG feature is hard to be paralleled for lack of feature points. The reason may lie in overlapping cell. Statistical histograms of gradient directions are very easy to generate conflicts in memory accesses of computing units. While using parallel SIFT features, because of the need to obtain images pyramid, it will not save computational overhead but will lower the performance of the algorithm when the feature points' size is small. Therefore, the main work of GPU acceleration is summarized as follows: (1) picking up the suitable image feature extraction algorithm which is easy to be parallelized and robust under small scale data. (2) By taking benefits of mobile phone GPU, parallelize the feature extraction in face alignment problem and the calculation of SDM algorithm

For the judgment of local description features, we derive heuristics from the linear discrimination analysis (LDA). Given an algorithm, denoted as A_i, we expected local features extracted using A_i should have smaller covariance at the same feature point in training set D. The features extracted at different feature points should have enough difference. Apply the algorithmic idea of LDA; that is, in the same class, variance of features should be as small as possible. On the contrary, the variance between classes should be as large as possible. We denote $L_{i,j}^k$ as the local feature extracted in the k_{th} sample's j_{th} feature points P_j using algorithm A_i. The covariance of P_j's local feature can be expressed as

$$\operatorname{Cov}(i,j) = \frac{\sum_k \left(L_{i,j}^k - m_{i,j}\right)\left(L_{i,j}^k - m_{i,j}\right)^T}{|D|}, \tag{17}$$

where $m_{i,j}$ is the mean of all samples feature extracted at P_j using algorithm A_i, $m_{i,j} = \sum_k L_{i,j}^k/n$.

Thus, the covariance of different feature points using A_i can be calculated with

$$\operatorname{Cov}(i) = \frac{\sum_j \left(m_{i,j} - \overline{m}_i\right)\left(m_{i,j} - \overline{m}_i\right)^T}{P}, \tag{18}$$

where $\overline{m}_i = \sum_j m_{i,j}/P$. We believe that if the feature extraction algorithm A_i is good enough, the covariance should be small with the same feature points but distinguishable at different feature points. Thus, we want to evaluate A_i using the similar method as used in LDA. Despite the Eigen vectors, we are only concerned about the portion of the max k Eigen values:

$$\operatorname{Score}\left(A_i\right) = \frac{\max_k \operatorname{Eigen}\left(\sum \operatorname{Cov}(i,j)\right)^{-1} \operatorname{Cov}(i)}{\sum \operatorname{Eigen}\left(\sum \operatorname{Cov}(i,j)\right)^{-1} \operatorname{Cov}(i)}. \tag{19}$$

Finally, we choose the algorithm with largest portion of largest k Eigen values:

$$A^* = \arg\max_i \operatorname{Score}_i\left(A_i\right). \tag{20}$$

We have analyzed recent popular local feature extraction algorithms for computer vision, including SIFT, SURF [28], HOG, BRIEF [29], ORB [30], BRISK [31], Freak [32], and MRLBP [27]. And we discovered that Freak has better performance in the problem of face alignment. The experiment details will be discussed in Section 5.

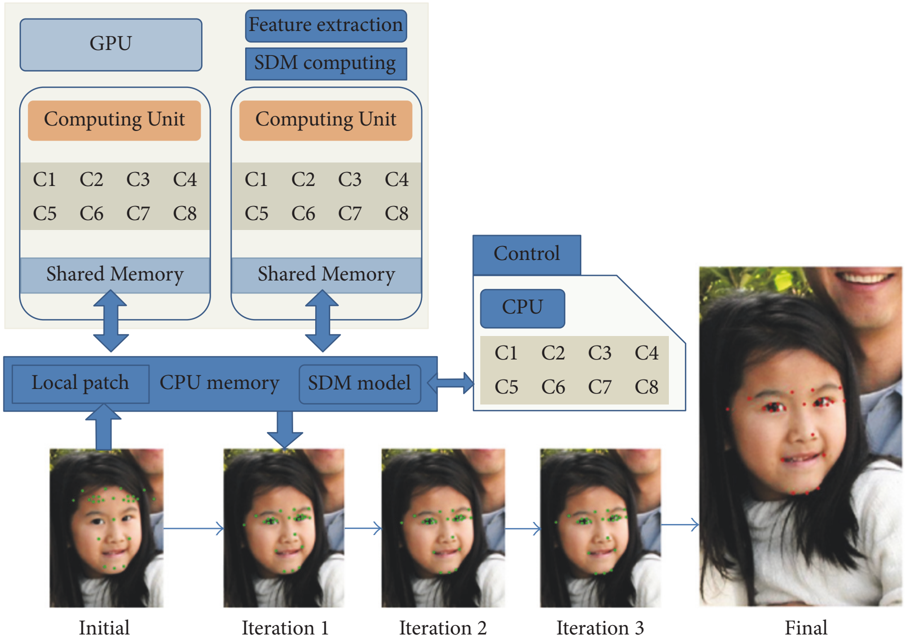


Figure 3: GPU parallel model for fusion feature extraction and SDM.

SDM face alignment algorithm does not have sufficient data to apply GPU acceleration. And if we do so, we will make both GPU-shared memory and CPU memory frequently visited. This will not only affect the performance of the algorithm, but also reduce memory life. Therefore, facilitating GPU architecture of mobile phone, we design a parallel SDM and feature extraction mode. As shown in Figure 3, our method parallelized features extraction and model solution of SDM. First, we load the pixel values of the subblocks at each feature point in memory according to the location of each feature point in the initial shape. Please note that if the pixel values in the subgraph block need to be accessed k times, we will store k copies of the pixel values in the shared memory. At the same time, the SDM algorithm is applied to the model fractions and image features at each patch, which is stored in the shared memory. Once the feature extraction calculation of a feature point is completed, the component of the image feature at the feature point is directly taken as the contribution to the overall shape regression.

5. Experiment Results & Discussion

In our experiment setup, we firstly evaluate the performance of local features as described in Section 4 with $k = 30$. In Table 1, we denote (Sep.) as the separable score according to (20), (p/s) as the numbers of feature points can be processed by these algorithms. To evaluate performance in mobile terminal, we use Qualcomm snapdragon 800 series platform. The dataset we used is a combination of LFW66 [33] and Helen [34].

Table 1: Performance comparison of each feature.

P.	A.							
	Sift	Hog	LBP	BRIEF	BRISK	ORB	FREAK	MRLBP
Sep.	0.6	0.67	0.4	0.47	0.44	0.45	0.65	0.62
p/s	47	66	200	131	147	112	150	132

As we can see in Table 1, the traditional local feature algorithm still has certain advantage regarding robustness or running speed. The most stable feature is still the HOG feature, but the fastest algorithm is the LBP feature. Considering the stability and running speed of the feature, the fast retinal feature points (Freak) proposed in recent years has more practical value.

Consequently, we compare the performance of the original SDM algorithm with our optimization on the aspects of efficiency and robustness. In order to prove the effectiveness of each optimization step, the whole optimization procedure is divided into C-SDM: the clustering optimization based on the feature point correlation; PC-SDM: the dimensional reduction optimization based on principal component analysis and feature point clustering; GPC-SDM: the face alignment algorithm which is finally proposed in this paper with adding dynamic GPU parallel optimization. First, we compare the optimization algorithms of each step with the original SDM. In order to quantify the problem, we first define the accuracy of the face regression algorithm with

$$p\left(I_k\right) = \frac{\sum_p \left\|\hat{y}_k^p - y_k^p\right\|}{P\left\|y_k\left(l\right) - y_k\left(r\right)\right\|}, \tag{21}$$

where I_k is the k_{th} image in testing set and $\hat{y}$ represents the estimated shape. The denominator is the distance between left

Table 2: Performance comparison with state-of-the-art algorithms.

	SDM-SVR	SDM(L1)	PC-SDM	GPC-SDM	EST	F-LBF	F-cGPRT
Fps.	4.5	47.5	27.5	48.6	100	150	220
Success.	0.92	0.85	0.915	0.906	0.72	0.75	0.78

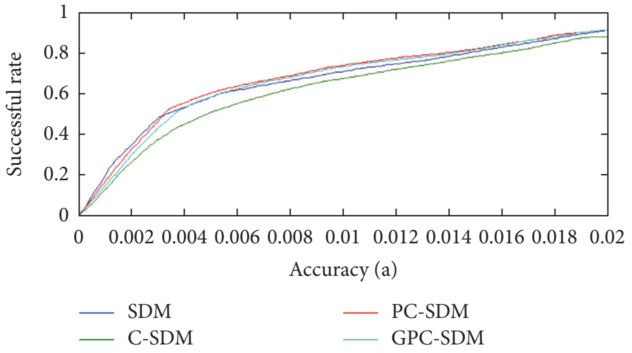


Figure 4: The plot of success rates.

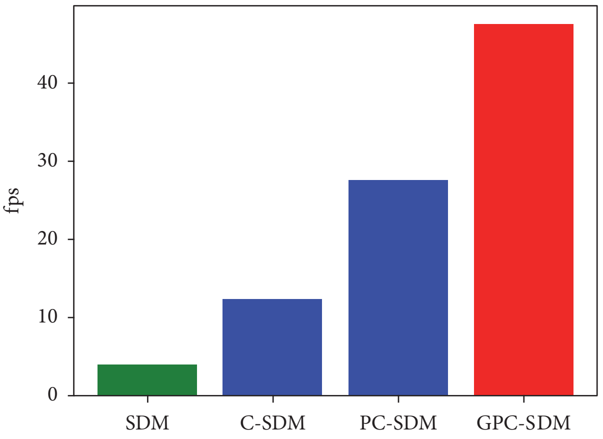


Figure 5: The tested fps of each optimization step.

and right eye, which is used to normalize the estimation error. We define the successful score of certain precision as follows:

$$\text{success}(\alpha) = \frac{\sum_k \delta(p(I_k) \le \alpha)}{C}, \qquad (22)$$

where δ is the indicator pointing out whether the testing sample's estimation error is lower than α and C is the number of testing samples. The successful rate of different optimization with growing α is shown as Figure 4.

Combined with Figures 4 and 5, we can see that each phase of the optimization has significantly improved against the previous phase. Compared with the original SDM algorithm, C-SDM algorithm has a slight decline in the success rate. PC-SDM algorithm, integrating PCA and C-SDM, can enhance the performance further without affecting the success rate. Though the GPU-based GPC-SDM optimization should have the same success rate curve with the PC-SDM algorithm, due to the excessive GPU floating-point truncation error, the success rate decreased slightly in high precision conditions. For all the algorithms, a relaxation of the accuracy condition can lead to the same success rate as the SDM algorithm. In a word, through comparison experiment, we can see that the optimization scheme proposed in this paper can achieve significant improvement in performance.

Finally, we compare our last two modifications with the state-of-the-art face feature points detection methods with speed advantage. The experiments were cross datasets designed to prevent overfitting phenomenon; that is, if the model was trained on one dataset, saying the Helen dataset, it must be evaluated with other datasets, say the LFW. Thus, the precisions may slightly different from the publications of the original works. Also, the fps decreases dramatically since the experiment environment moves to mobile terminal. To simplify the experiments, we choose $\alpha = 0.02$ and compare both fps and success rate with state-of-the-art algorithms: the ensemble regression trees (EST) [9], the Gaussian process regression trees (F-cGPRT) [10], the local binary feature regression (F-LBF) [11], and the L1 norm penalty SDM (SDM(L1)) [12].

As shown in Table 2, the performance of PC-SDM and GPC-SDM boost the speed of face feature points detection and preserve the successful rate of the most robust SDM algorithm. Other state-of-the-art algorithms used regression trees or forest and simple image features to improve the speed. However, these approaches do not break through the computation framework proposed by SDM. And nowadays regression trees algorithms are easy to overfit the training set. This can be easily observed in Table 2 where although their approaches were announced to be precise, the successful rate downgraded with cross datasets evaluation. One successful modification of SDM is SDM(L1) [12], which employs sparsity with the SDM models and can speed up the algorithm easily and significantly. However, it also sacrifices the success rate compare to original SDM algorithm with support vector regression. Thus, although our proposal did not achieve the fastest performance even with GPU acceleration, it has more practical value considering both robustness and real-time requirements.

6. Conclusion

In this paper, we transplant the SDM algorithm on mobile devices with the following work: first, based on the correlation analysis between the feature points on the statistics of face shapes, we introduce spectral cluster and PCA to reduce the model size of SDM. Second, a local feature evaluation method based on linear discrimination classification is proposed, which decouples feature evaluation and face alignment algorithm to facilitate the experimental work. Third, a GPU-based SDM model is proposed, which combines feature extraction and model solving. Finally, the potential of SDM algorithm is fully exploited, and the SDM algorithm is optimized and transplanted to mobile terminal.

Face alignment algorithm has broad application prospects, but the research of this problem is still in its initial stage. We still need to improve the success rate of the face alignment algorithm, especially in the presence of occlusion, light changes, night light, and face 3D pose changes. The algorithm of face alignment and facial feature point tracking success rate is still insufficient. Studying how to overcome these problems is important in our next research work.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] Z. Ning, X. Hu, Z. Chen et al., “A cooperative quality-aware service access system for social internet of vehicles,” *IEEE Internet of Things Journal*, pp. 1-1, 2017.

[2] X. Hu, T. H. S. Chu, V. C. M. Leung, E. C.-H. Ngai, P. Kruchten, and H. C. B. Chan, *A Survey on Mobile Social Networks: Applications, Platforms, System Architectures, and Future Research Directions , IEEE Communications Surveys Tutorials*, vol. 17, A Survey on Mobile Social Networks, Applications, 2015.

[3] X. Hu, T. H. S. Chu, H. C. B. Chan, and V. C. M. Leung, “Vita: a crowdsensing-oriented mobile cyber-physical system,” *IEEE Transactions on Emerging Topics in Computing*, vol. 1, no. 1, pp. 148–165, 2013.

[4] X. Hu, X. Li, E. C.-H. Ngai, V. C. M. Leung, and P. Kruchten, “Multidimensional context-aware social network architecture for mobile crowdsensing,” *IEEE Communications Magazine*, vol. 52, no. 6, pp. 78–87, 2014.

[5] Y. Heng et al., *An Empirical Study of Recent Face Alignment Methods*, Computer Science, 2015.

[6] X. Xiong and F. De la Torre, “Global supervised descent method,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR ’15)*, pp. 2664–2673, IEEE, Boston, Mass, USA, June 2015.

[7] S. Zhu, C. Li, C. C. Loy, and X. Tang, “Face alignment by coarse-to-fine shape searching,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015*, pp. 4998–5006, USA, June 2015.

[8] V. Kazemi and J. Sullivan, “One millisecond face alignment with an ensemble of regression trees,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR ’14)*, pp. 1867–1874, Columbus, Ohio, USA, June 2014.

[9] X. Cao, Y. Wei, F. Wen, and J. Sun, “Face alignment by Explicit Shape Regression,” in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2012*, pp. 2887–2894, USA, June 2012.

[10] D. Lee, H. Park, and C. D. Yoo, “Face alignment using cascade Gaussian process regression trees,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015*, pp. 4204–4212, USA, June 2015.

[11] S. Ren, X. Cao, Y. Wei, and J. Sun, “Face alignment at 3000 FPS via regressing local binary features,” in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014*, pp. 1685–1692, usa, June 2014.

[12] Q. Liu, J. Deng, and D. Tao, “Dual sparse constrained cascade regression for robust face alignment,” *IEEE Transactions on Image Processing*, vol. 25, no. 2, pp. 700–712, 2016.

[13] A. Jourabloo and X. Liu, “Large-pose face alignment via CNN-based dense 3D model fitting,” in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, pp. 4188–4196, usa, July 2016.

[14] T. Baltrusaitis, P. Robinson, and L.-P. Morency, “Constrained local neural fields for robust facial landmark detection in the wild,” in *Proceedings of the 14th IEEE International Conference on Computer Vision Workshops (ICCVW ’13)*, pp. 354–361, Sydney, Australia, December 2013.

[15] M. K. Hasan, C. Pal, and S. Moalem, “Localizing facial keypoints with global descriptor search, neighbour alignment and locally linear models,” in *Proceedings of the 2013 14th IEEE International Conference on Computer Vision Workshops, ICCVW 2013*, pp. 362–369, aus, December 2013.

[16] S. Jaiswal, T. R. Almaev, and M. F. Valstar, “Guided unsupervised learning of mode specific models for facial point detection in the wild,” in *Proceedings of the 2013 14th IEEE International Conference on Computer Vision Workshops, ICCVW 2013*, pp. 370–377, aus, December 2013.

[17] B. Martinez, M. F. Valstar, X. Binefa, and M. Pantic, “Local evidence aggregation for regression-based facial point detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 5, pp. 1149–1163, 2013.

[18] K. Choi, K.-A. Toh, and H. Byun, “Realtime training on mobile devices for face recognition applications,” *Pattern Recognition*, vol. 44, no. 2, pp. 386–400, 2011.

[19] P. A. Tresadern, M. C. Ionita, and T. F. Cootes, “Real-time facial feature tracking on a mobile device,” *International Journal of Computer Vision*, vol. 96, no. 3, pp. 280–289, 2012.

[20] W. Herman, *Fix-Point Method*, Encyclopedia of statistical sciences, 1983.

[21] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, “Active shape models—their training and application,” *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 38–59, 1995.

[22] D. Jian-kang, W. Can-tian, and L. Qing-shan, “Fast Face Alignment Method Based on Hierarchical Model and it’s Application on Mobile Device,” *Computer Science*, vol. 12, p. 65, 2014.

[23] H. jiang, *The research and implement of face feature points localization based on android platform*, Beijing Jiaotong University, 2013.

[24] W. L. W. Hp, “Real time face pose estimation based on 3D model implemented in android,” *International Journal of Computer Applications*, 2015.

[25] T. B. N. Dalal, “Histograms of oriented gradients for human detection,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR ’05)*, vol. 1, pp. 886–893, June 2005.

[26] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[27] L. Liu, S. Lao, P. W. Fieguth, Y. Guo, X. Wang et al., “Median robust extended local binary pattern for texture classification,” *IEEE Transactions on Image Processing*, vol. 25, no. 3, pp. 1368–1381, 2016.

[28] H. Bay, T. Tuytelaars, and L. Van Gool, “SURF: speeded up robust features,” in *Proceedings of the in European Conference on Computer Vision*, 2006.

[29] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, “BRIEF: binary robust independent elementary features,” in *Computer Vision—ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5–11, 2010, Proceedings, Part IV*, vol. 6314 of *Lecture Notes in Computer Science*, pp. 778–792, Springer, Berlin, Germany, 2010.

[30] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “ORB: an efficient alternative to SIFT or SURF,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV ’11)*, pp. 2564–2571, Barcelona, Spain, November 2011.

[31] S. Leutenegger, M. Chli, and R. Y. Siegwart, “BRISK: binary robust invariant scalable keypoints,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV ’11)*, pp. 2548–2555, Barcelona, Spain, November 2011.

[32] A. Alahi, R. Ortiz, and P. Vandergheynst, “FREAK: fast retina keypoint,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR ’12)*, pp. 510–517, June 2012.

[33] B. Huang Gary et al., *Labeled Faces in the Wild: A Database for Studying face recognition in unconstrained environment*, University of Massachusetts, Amherst, MA, USA, 2007.

[34] Le. Vuong et al., “Interactive Facial Feature Localization,” in *Proceedings of the European Conference on Computer Vision*, 2012.

Journal of Engineering
The Scientific World Journal
International Journal of Rotating Machinery
Journal of Sensors
Advances in Multimedia
Advances in Civil Engineering
Journal of Control Science and Engineering
Journal of Robotics
Journal of Electrical and Computer Engineering
Advances in OptoElectronics
VLSI Design
International Journal of Navigation and Observation
Modelling & Simulation in Engineering
International Journal of Aerospace Engineering
International Journal of Chemical Engineering
International Journal of Antennas and Propagation
Active and Passive Electronic Components
Shock and Vibration
Advances in Acoustics and Vibration