Appl. Math. Inf. Sci. 6 No. 2S pp. 403S-408S (2012)

Applied Mathematics & Information Sciences
An International Journal

Robust Facial Expression Recognition Using a Smartphone Working against Illumination Variation

Kyoung-Sic Cho[1], In-Ho Choi[1] and Yong-Guk Kim[1]
[1] Department of Computer Engineering, Sejong University, 98 Kunja-Dong, Kwangjin-Gu, Seoul, Korea
Corresponding author: Email: ykim@sejong.ac.kr

Received June 22, 2010; Revised March 21, 2011; Accepted 11 June 2011
Published online: 1 January 2012

Abstract: Active Appearance Model (AAM) [1, 4] has been favoured among computer vision researchers, since it is useful for tracking human face or diverse objects. Moreover, recent development of the inverse composition method allows us to relieve computational burden in tracking human face with the real-time basis. The progress of fast algorithm development makes it possible that we may be able to track a human face within video image feeding from a smartphone. However, such mobile device gives us some limitation in computing power, but also illumination environment around it is not favourable to vision algorithm designers. This paper presents new method by which such illumination variation occurred in the mobile environment can be overcome. We have found that the Difference of Gaussian (DoG) kernel preceded the AAM stage is very effective in tracking human face including important facial features such as eyes, nose and mouth, despite a strong directional illumination. Performance of the proposed system is evaluated for diverse illumination conditions, and result suggests that the DoG kernel, that has been identified as biologically plausible, can plays an important role against such odd illumination environment. This algorithm has been implemented on a smartphone for the purpose of human facial expression recognition and it works well with a camera as an application program.

Keywords: Active Appearance Model, Difference of Gaussian, Facial Features, Facial Expression, Smartphone, Application Program

1 Introduction

Facial expression plays an important role in recognizing emotional state of a human being. Recent advancement of computer algorithms and computing power make it possible to recognize human facial expressions using a computer. Indeed, facial expression recognition using a computer has been used in many areas such as entertainment, vision interface, psychiatry and so on. We typically assume that the number of facial expressions generated by human face could be thousands. However, according to previous studies, humans' basic facial expressions can be divided into six categories (surprise, fear, sadness, angry, disgust, and happiness), although the human face can express many other different facial states [2]. Since AAM is one of the top-down vision algorithms, it requires training session, in this case tracking human facial expressions, using a lot of facial expression images. The theory of six basic facial expressions proposed by Ekman provides a useful ground work for our purpose.

To recognize facial expression using a computer, a lot of computing resources are required. Yet, most of the facial expression recognition by machine has been demonstrated using a desktop computer. In this paper, we show that a smartphone is a reasonable platform on which a bundle of facial expression recognition algorithms and others can be implemented, as the computing performance of the present smartphone has been improved substantially. The required algorithms for this

purpose include face detection, facial feature tracking, and classification of facial expressions.
In this study, we are going to describe a robust facial expression recognition system on a smartphone. First, the basic theory of AAM [1, 4] is described and then the performance between several edge-detector algorithms related with AAM is presented. Secondly, for the purpose of classification of facial expressions, the back-propagation neural network is used. The experiment and result will be given. Finally, implementation of these algorithms in a smartphone is presented, and conclusion and future work will be discussed.

2 Active Appearance Model

AAM is a flexible model, which is for modeling a face. It basically consists of the shape and the appearance, and they are used to synthesize a model similar to the object's image such as a face. In order to synthesize a face, we need to find the model parameters using an image alignment algorithm. Recently, the Inverse Compositional Image alignment (ICIA) method has been developed to reduce the computing time for such image alignment task [3, 5], which makes it possible to track the face even in a real-time basis. Below is a brief outline of AAM [4].

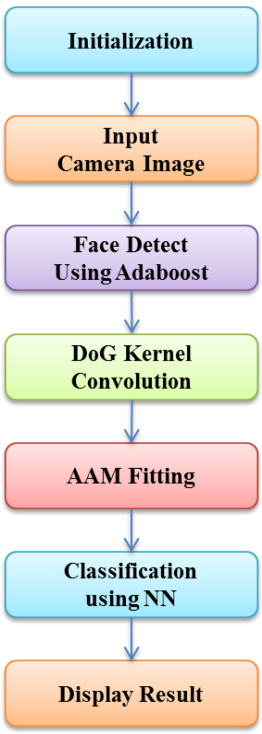


Figure 1: The system flow of robust facial expression recognition on a smart phone.

First of all, the shape of an AAM is defined by mesh generated by connecting landmarks, which are manually annotated on a face image. Mathematically, the shape is defined using a vector.

$$s = \{x_1, y_1, x_2, y_2, \ldots, x_v, y_v\}^{\mathrm{T}} \tag{2.1}$$

The shape vector allows us linear variation of the vector. Using such manipulation, we can define it as follow:

$$s = s_0 + \sum_{i=1}^{n} p_i s_i \tag{2.2}$$

In equation (2.2), coefficient p_i is the parameter of the shape, s_i is the orthornormal vectors that are obtained by Principal Components Analysis (PCA), and s_0 is the mean shape.

Secondly, the appearance of the AAM, in other words the color, is defined by the pixel x =(x,y)T within the shape s, warp to the mean shape s0. Similar to the shape case, the appearance can be defined by linear variation as follow:

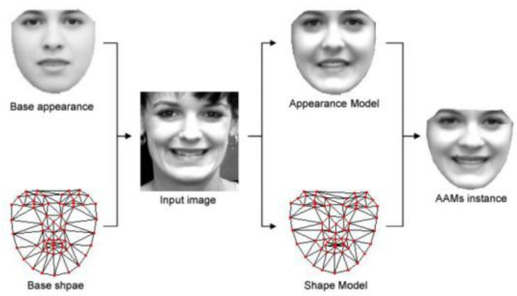


Figure 2: Generation of an AAM instance for an input image using both the shape and the appearance models.

$$A(\mathrm{x}) = A_0(\mathrm{x}) + \sum_{i=1}^{m} \lambda_i A_i(\mathrm{x}) \tag{2.3}$$

In equation (2.3), the coefficient λ_i is the appearance parameter, A_i is the orthornormal vectors that are also obtained by PCA method. A_0 is the mean appearance.

$$\mathrm{M}(W(\mathrm{x};\, \mathrm{p})) = A(\mathrm{x}) \tag{2.4}$$

In this equation (2.4), M represents the model instance. The model instance can be generated by warping the pixel x within the shape s by updating the shape s using parameter p. Here, the parameter p is calculated using the ICIA method [2].

$$\sum_x \left[I(W(\mathrm{x};\mathrm{p}) - A_0(W(\mathrm{x};0))) - \nabla A_0 \frac{\partial W}{\partial \mathrm{p}} \Delta \mathrm{p} \right]^T \qquad (2.5)$$

$$\Delta \mathrm{p} = \mathrm{H}^{-1} \sum_x \left[\nabla A_0 \frac{\partial W}{\partial \mathrm{p}} \right] \left[I(W(\mathrm{x};\mathrm{p})) - A_0(\mathrm{x}) \right] \qquad (2.6)$$

In equation (2.5), $W(\mathrm{x};0)$ is supposed to be the identity warp, $\Delta \mathrm{p}$ is calculated by the linear square method in equation (2.6). In that equation, $\nabla A_0 \frac{\partial W}{\partial \mathrm{p}}$ is a steepest descent image, which is obtained by gradient ∇A_0 and Jacobian $\frac{\partial W}{\partial \mathrm{p}}$. We calculate Gradient ∇A_0 on A_0 using the shape vectors and warp parameter[2], obtain Jacobian $\frac{\partial W}{\partial \mathrm{p}}$. H^{-1} is an inverse Hessian matrix, which is calculated by the steepest descent image and it is transposed as $\mathrm{H} = \sum_x \left[\nabla A_0 \frac{\partial W}{\partial \mathrm{p}} \right]^T \left[\nabla A_0 \frac{\partial W}{\partial \mathrm{p}} \right]$. Fig. 2 illustrates the procedure of generating a model instance for an arbitrary input image.

3 DoG kernel proceeded the AAM stage

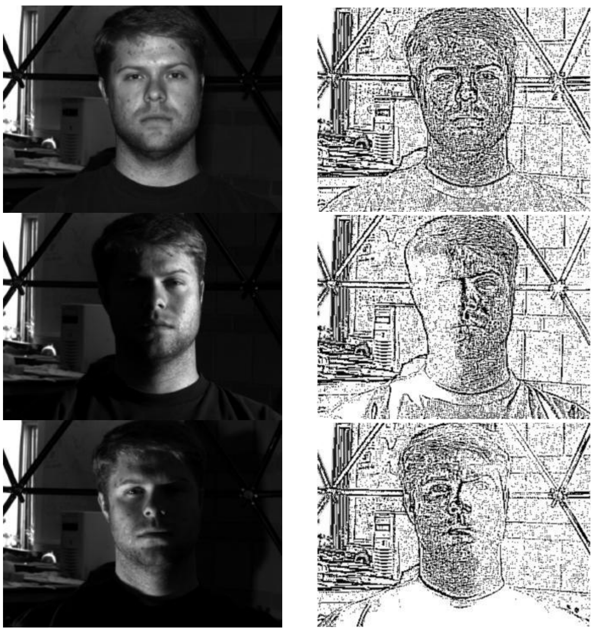


Figure 3: The left shows the original images and the right the convoluted images by DoG kernel. The σ value of DoG kernel is 0.3 and 0.6, respectively.

Although AAM is known to be effective in modeling the face or facial expression, the facial features in the face are prone to vary by the illumination direction change or an occlusion by face pose variation. Since the mobile illumination environment is normally unstable, it is not surprising that the conventional AAM often fails to converge during the fitting process. We thought that an edge detector could be useful if it is proceeded the AAM stage, since edge detection means to discriminate facial features such as eyes, mouth, nose from the remaining areas within in a face. Among many well-known edge detection algorithms, we have tested the Difference of Gaussian (DoG) kernel and Canny edge detector.

First, a DoG kernel consists of two Gaussian kernels, which have different standard deviations. We can obtain a DoG convolution image by convolving two Gaussian kernels to the given image. The DoG kernel can be defined as follow:

$$\begin{aligned} D(x,y,\sigma) &= (G(x,y,\sigma) - G(x,y,\sigma)) \\ &\quad * I(x,y) \\ &= L(x,y,k\sigma) - L(x,y,\sigma) \end{aligned} \qquad (3.1)$$

In equation (3.1), $L(x,y,k\sigma)$ and $L(x,y,\sigma)$ are Gaussian kernels, that have different standard deviation. Fig. 3 compares the input images from Cohn-Kanade facial expression database [6] to the convoluted images using the DOG kernel.

Secondly, Canny detector is best known as an optimum edge detection algorithm, by which any features within an image can be extracted using a Sobel operator, after applying Gaussian kernel to the image. The task of this algorithm is to extract smooth edges by removing non-necessary features. The top images in Fig. 4 compare the visual characteristics of three cases: the original image in the left, the DoG convolved image in the middle and Canny edge detector in the right. Notice that the facial features of the original face have been more accentuated within the DoG case than those in the Canny detector case. As the main purpose of the facial expression modeling is to track the facial features correctly, enhancement of facial feature using the DoG kernel could help in modeling the face using the AAM, since the facial features are prone to disappear or blur when the illumination is strongly directional or some occlusion occurs by face pose variation. On the other hand, the image Canny detector case (right) contains rather smooth edges without strong facial features. In the bottom of Fig. 4, the system flow of the AAM with the DoG kernel case is illustrated.

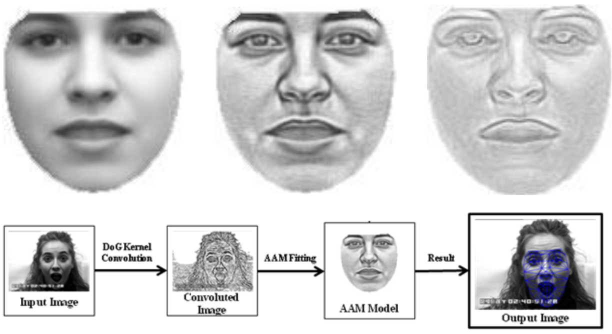


Figure 4: Appearance models for three cases (Left: the input image, Middle: DoG kernel, Right: Canny edge detector) (Top). The system flow of AAM with DoG kernel (Bottom).

4 Experiments and Result

The aim of the first experiment is to evaluate how new kernel installation before the AAM stage affects AAM fitting performance. Among many standard face databases, since Yale face database is well known for its diverse face images acquired by varying the illumination direction as well as pose systemically, we have adopted it for our illumination variation test [9].

Figure 5: AAM fitting for the original AAM (Top), the DoG kernel (Middle) and Canny edge (Bottom) using Yale face database to evaluate performance against diverse illuminations.

Fig. 5 shows three different AAM fitting images: the original AAM (top), DoG & AAM (middle), Canny & AAM (bottom). Notice the difference between the original case and two kernel cases. On the top of Fig. 5, two AAM templates are either elongated or shrank especially around the dark side of the face, whereas such elongation or shrinkage is not obvious within the middle and the bottom images. In Fig. 6, we depict the result by accumulating the errors for different AAM conditions. It is clear that the DoG AAM outperforms the Canny AAM and the original AAM. Notice that the original AAM makes a huge error about 10 when it does not fit to the face.

In Table 1, we summarize result of the performance test. The mean error for the DoG & AAM case is smaller than that for two other cases. Notice that the mean error of the original AAM is about twice than that of the other two cases.

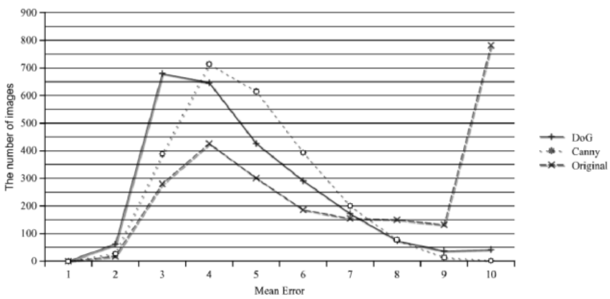


Figure 6: Comparison curves of making errors for three different AAM conditions.

	Mean Error	*Standard Deviation*
Original AAM	7.57294	4.78771
DoG + AAM	4.11276	1.71121
Canny + AAM	4.29566	1.32626

Table 1. Mean and Standard Deviation(SD) between ground truth and fitted data.

For the second experiment, we have employed a standard Cohn-Kanade facial expression database from Carnegie-Mellon University for training AAM model. All images are collected under the structured illumination by asking subjects to make facial expressions [6]. In the present experiment, 252 images among them are used for facial expression recognition test. We have employed a neural network for classifying facial expressions [7, 8]. Table 2 shows the result from our recognition experiment. The test set consists of sadness (54), surprise (64), happiness (69), and neutral (82) cases. This confusion matrix illustrates how our AAM model recognizes correctly as well as makes errors for each facial expression. For instance, our model makes 6 mistakes as happiness and 4 mistakes as neutral emotion, respectively, for the surprise images as indicated in the 3rd row in the table.

	Happiness	Neutral	Surprise	Sadness
Happiness	66	1	0	2
Neutral	11	64	0	7
Surprise	6	4	52	2
Sadness	9	4	2	39

Table 2. Confusion matrix for the facial expression recognition test

In order to implement a real-time facial expression recognition system on a smartphone, an iPhone 4, that has 1GHz CPU with iOS 4 and Xcode 4.1 compiler, is used for this study. Given that it is one of many embedded systems and computing speed is an important factor, we have measured the processing speed for this task. In Table.3, the processing times for the facial expression recognition of the original AAM and the AAM with DoG kernel case are compared. Note that the speed difference between two cases is not significant. Result indicates that the facial expression recognition task with AAM using DoG kernel can be carried out about 3 times per second, suggesting that a smartphone is a reasonable platform for the real-time facial expression recognition system. [10] Four image shots captured from our iPhone 4 are shown in Fig. 7, in which the top images are taken during the original AAM and the bottom images during the AAM with DoG kernel running, respectively.

	Original AAM (seconds)	AAM with DoG (second)
Model Instance	0.1858324	0.25154745
Recognition	0.063686813	0.056849313

Table 3. The mean processing time required for the facial expression recognition task.

Here, you may be able to identify how each model accurately fits the given facial expression, as the AAM template is overlaid on input image. For instance, the template in the top-right case does not align with the outline of the face well, whereas it fits the face well in the bottom-right image. At the bottom of each image, an emoticon represents the emotional state recognized by the AAM. Notice that the AAM with DoG recognizes subject's facial states better than the original AAM does as two emoticons in the top-right is compared with in the bottom-right. User can choose either emoticon or written description of the facial expression by sliding an emoticon button. Similarly, the AAM template overlaid on the face area can be removed by sliding a warp button.

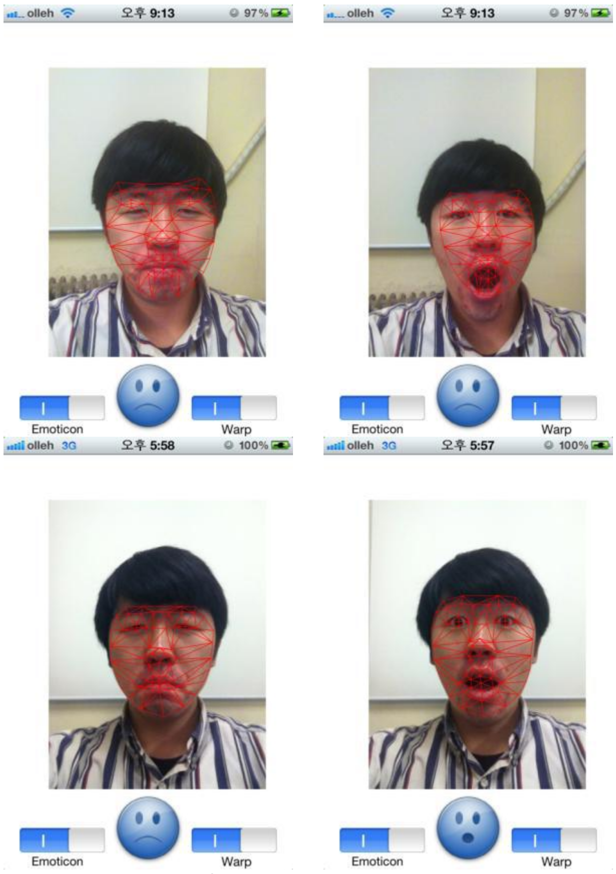


Figure 7: Real-time facial expression recognition system on a smartphone.

6 Conclusions

Among recent advancement of the smartphone applications, Augmented Reality (AR) is becoming an important area. Computer vision techniques provide essential tools, such as finding and tracking some objects within the images feeding from the camera. However, illumination conditions around the mobile phone are not favorable to carry out such tasks, since the illumination is often strongly directional or dim. In this study, we propose new method by which it sidesteps such unfavorable illumination conditions.

It is found that a DOG kernel preceding the AAM model appears to be very effective to overcome dark shadow or occlusion in tracking the face robustly. We have tested this algorithm for diverse illumination conditions using Yale face databases. Result suggests that fitting performance has been much improved. The processing speed of new AAM is almost similar to that of the original AAM in a smartphone. We expect that the present method can be applied to other tracking applications as well.

Acknowledgements

This research was supported by a grant “Creative Software Project” from NIPA (National IT Industry Promotion Agency).

References

[1] G. Edwards, C.J. Taylor, and T.F. Cootes. Interpreting Face Images Using Active Appearance Models. In Proc. International Conference on Automatic Face and Gesture Recognition, June (1998), 300-305.

[2] P. Ekman. Universal and cultural differences in facial expressions of emotion. In J Cole (Ed) Nebraska symposium on motivation 1971 (pp 207-283) Lincoln: University of Nebraska Press (1972)

[3] I. Matthews and S. Baker. Active Appearance Models Revisited. International Journal of Computer Vision, (2004), 135-164.

[4] G. Xinbo, S. Ya, L. Xuelong, and T. Dacheng. A review of Active Appearance Models. IEEE trans. on Sys., Man, and Cyber., vol. 40, No. 2, March (2010), 145-158.

[5] R. Gross, I. Matthews, and S. Baker. Constructing and fitting active appearance models with occlusion. proceeding of the IEEE Workshop on face processing in Video, June (2004).

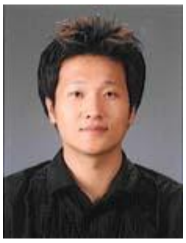
[6] T. Kanade, J. Cohn, and Y. Tian. Comprehensive database for facial expression analysis. In Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition. (2000), 46-53.

[7] C. L. Lisetti and D. E. Rumelhart. Facial expression recognition using a neural network. In Proceedings of the 11th International Florida Artificial Intelligence Research Society Conference (FLAIRS’98), Menlo Park, CA, (1998).

[8] E. T. Rolls and A. Treves. Neural Networks and Brain Function. Oxford, (1998).

[9] K.C. Lee and J. Ho and D. Kriegman. Acquiring Linear Subspaces for Face Recognition under Variable Lighting IEEE Trans. Pattern Anal. Mach. Intelligence, (2005), 27, 5 684-698

[10] P.A. Tresadern, et al., Real-time facial feature tracking on a mobile device, Int. Journal of Computer Vision, (2011) May



Cho Kyoung-Sic received his PhD from Sejong University, in the Department of Computer Engineering, Seoul, Korea. He works as a researcher in Hyosung-Nautilus Company. His research interests are in pattern recognition, facial expression recognition, and Human-Computer Interaction.

Choi In-Ho received his Master Degree from Sejong University and is currently a PhD candidate in the Department of Computer Engineering. His research interests are image processing, facial expression synthesis, and smartphone applications.

Kim Yong-Guk received his PhD degree from Cambridge University. He is currently an Associate Professor with the Department of Computer Engineering and Director of the Smartphone & Application Center at Sejong University, Seoul, Korea. His research interests are in the areas of facial expression recognition and synthesis, computational vision, and human-computer interaction. E-mail: ykim@sejong.ac.kr