Key Frame Extraction from Video Sequence Using a Face Quality Index Applied to Surveillance System

Sofina Yakhu and Nikom Suvonvorn*
Department of Computer Engineering, Faculty of Engineering, Prince of Songkla University,
Hat Yai, Songkhla 90112
E-mail: s4810167@psu.ac.th, kom@coe.psu.ac.th*

Abstract

We propose in this paper a method for measuring the quality of face images, detected from video sequence, which is used for key frame extraction. The method is aim for applying to the domain of video surveillance application. The inspection of covariance matrix values of face images within periods of time is the fundamental concept of our face quality index measurement. The system performance evaluation is done on testing dataset with discussion.

Keywords: Key Frame Extraction, Face Detection, Face Tracking, Face Quality Index, Video Surveillance

1. Introduction

In the domain of video surveillance system, the key frame extraction is very important issues. Many works have been published [1] [2]. Our goal on this research is inspired by the real problem. It's quite very often that the video obtained from CCTV system cannot indentify the person because of the low resolution or low image quality degraded by the compression. Although the current IP-camera using in the market is really advance with very high resolution (1-2M pixels) but in the real world application with a lot of cameras it is limited to the network bandwidth that cannot stream the full quality to the video network recorder. The usual resolution in general is VGA or CIE with MPEG4 or H.264 encoding.

The idea of our research is firstly try to identify the key frame from the low resolution and low quality video sequence streaming from the IP-camera. If there is an interesting event indicating by the key frame, the video network server will ask the camera to send the highest resolution of the current image frame. With this solution, we will obtain the best quality of image at a specific event while reducing network bandwidth. In this paper, we focalize only on the video analysis technique for key frame selection but not the overall system implementation. Our proposed method is emphasized on the face quality index measurement based on the covariance matrix values of detected face images.

The paper is organized as follows: section 2 reviews the face detection and tracking method, section 3 is devoted to our proposed method, section 4 presents the experimentation results with discussion, and the last sections summarized by conclusion.

2. Reviewed Techniques

In this section two algorithms for object detection and tracking are presented: the Harr-Like and Camshift algorithm respectively. We apply these methods as the initialization step of our method in the section 3.

2.1 Haar-Like Object Detection

The object detection techniques based on Harr-like features are proposed by several authors. In this paper we emphasize on the method of Viola and all [3][5]. The detection framework can be resumed into two important steps. Firstly, with a given over-complete set of Haar-like feature (180,000 rectangle features associated with each images sub-window) and a training set of positive and negative images, the weak AdaBoost learning algorithm is used both for selecting a small set of features and training the classifier. The algorithm is designed to select a single feature which best separates the positive from the negative examples. The classification function $h_j(x)$ consists of a feature $f_j(x)$, a threshold θ_j and p_j indicating the direction of the inequality sign. Here, x is a 24x24 pixel of an image.

$$h_j(x) = \begin{cases} 1 & \text{if} \quad p_j f_j(x) < p_j \theta_j \\ 0 & \text{otherwise} \end{cases}$$

Secondly, the classifiers (stages) are constructed into a cascade which is again trained using AdaBoost algorithm by adjusting the threshold to minimize false negatives. With the design, a positive result from the first stage triggers the better evaluation of second one while a negative result will be rejected immediately from the decision. This kind of decision tree increases the detection rate and inversely reduces the computation time.

2.2 Camshift Object Tracking

Camshift [4] is an object tracking algorithm derived from mean shift algorithm which is a non-parametric technique. The colored object represented in a probability distribution is tracked in the video sequence by climbing the gradient values in order to find the nearest dominant peak. Camshift tracks the object by updating its probability distribution dynamically which allows better tracking the moving object having size changing. The algorithm can resume by the following. For each video frame, the raw image is converted to a color probability distribution image via a color histogram model such as HSV space. The center and size of the tracking object are computed from the color probability image as its centroid and area. The current size and location of the tracked object are reported and used to set the size and location of the search window in the next video frame. The process is then repeated for continuous tracking.

3. Proposed Method

In this section we describe our proposed method in details how to select the key frame from video sequence using a face quality index. Our technique consist of two steps: face detection and tracking, and face quality index measurement.

3.1 Face Detection and Tracking

The aim of this step is to find the face area in each frame of video sequence that will be used further in the face quality measurement step. We combine two methods based on the work of Viola [3] for face detection and Bradski [4] for face tracking respectively. The face detection technique using Harr-like method achieves high detection accuracy while minimizing the computation time, tested with MIT+CMU frontal face test set. However, in the real situation for the video sequence from CCTV system where the resolution of images are large (normally VGA or CIF) and degraded by the compression, the detection rate decrease and false positive increase. Concerning the Camshift algorithm, the concept of tracking method operates by climbing the gradient of color probability distribution. In real situation, it is often that the background color or its texture is quite similar to the tracking region. As Camshift updates dynamically its probability distribution, this let to miss the tracking area with less accuracy in term of face localization.

We then join these methods in order to combine their main advantages that will increase relatively the detection rate and the robustness. The overall process (figure 1) is performed by the following: for each video frame, face detection by Haar-Like algorithm is applied. If a face is detected, its histogram in HSV model is computed and used as color probability distribution for Camshift initialization or correction. If no faces are detected in a frame, the tracked area obtained from Camshift is automatically assumed to be the missing face. In the correcting phase where the face is detected and the tracked face is found, the correction is done simply by a low-pass filter (average) on its size and position.

Figure 2 show some results from our detection and tracking approach. At the end of this step we obtained a sequence of face images extracted from every frame in the video sequence.

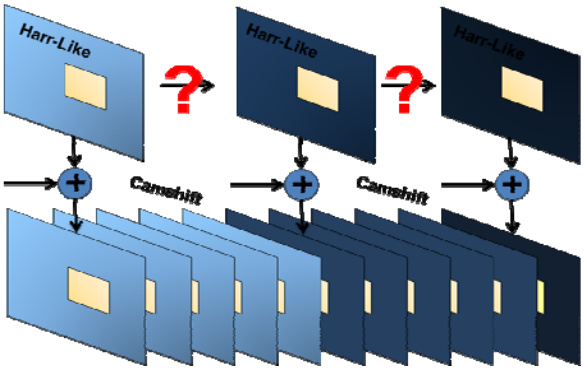


Figure 1 : Harr-Like and Camshift Mixture .

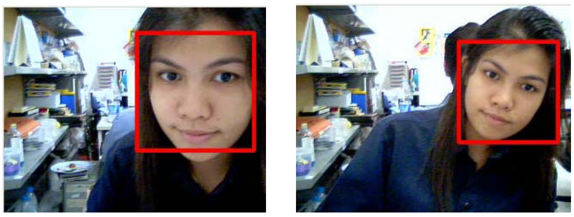


Figure 2. Results of Harr-like and Camshift Mixture

3.2 Face Quality Index for Key Frame Extraction

The face quality measurement is aim for determining the key frame from the video sequence which relate to the best appearance feature of face characteristics. The procedure can be divided into two stages: post-processing and face quality measurement. Figure 3 illustrate the overall process.

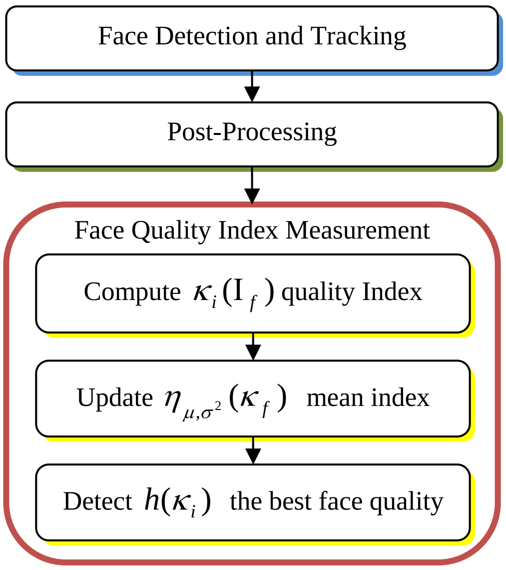


Figure 3. Face Quality Index Measurement Process

In the post-processing, the detected face images from every frame are transformed into the same size (150x150) with bilinear interpolation, and then the basic histogram equalization is applied. This aim to reduce the two noise factors, the number of pixels and the intensity variation effected from light changing in the environment, which will interfere to our face quality measurement method.

After the post-processing, the measurement of face quality can then be computed. Let $\kappa_i(\mathbf{I}_f)$ is the quality index of face image at frame$_i$. It is defined by the following equation:

$$\kappa_i(\mathbf{I}_f) = \mathrm{Trace}\big(\Sigma(\mathbf{I}_f)\big) + \mathrm{Trace}\big(\Sigma(\mathbf{I}_f^T)\big)$$

Here Σ is the covariance matrix of face image $\mathbf{I}_f$. This function gives a big value when the image is burred and a small one for very noisy image. However, the function indicates us only the tendency of values with respect to the noisy, good, and burred images. There is no the exact value that could be used to evaluate a best quality face. Actually, this value depend on the physical appearance at that time, such as the characteristics of moving object, the contrast changing in the environment, and etc. Intuitively, our method will takes the consideration on these parameters by looking globally the best face image in a short period of time. The procedure of our method is the following: we first compute a reference value and then determine the best quality face to those which have $\kappa_i(\mathbf{I}_f)$ value around that reference. We introduce the Gaussian model as low-pass filter η for determining the reference value μ which needs to be updated continuously.

$$\eta_{\mu,\sigma^2}(\kappa_i) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(\kappa_i-\mu)^2}{2\sigma^2}}$$

Then, the best quality face image can be detected by the following index function:

$$h(\kappa_i) = \begin{cases} 1 \quad \text{if} \quad \mu_{\kappa_i} + nm\sigma_{\kappa_i} < \kappa_i < \mu_{\kappa_i} + (n+1)m\sigma_{\kappa_i} \\ \qquad 0 \quad \text{otherwise} \end{cases}$$

So, the n and m are our model parameter representing the sub-window location referred to the mean value and the sub-window size respectively. These parameters are determined experimentally. We will discuss in the next section. Figure 4 shows the best face quality index $h(\kappa_i)$.

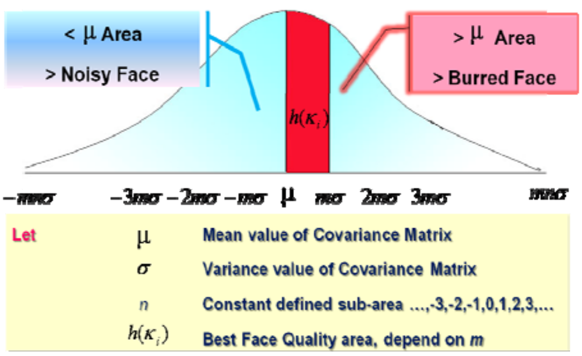


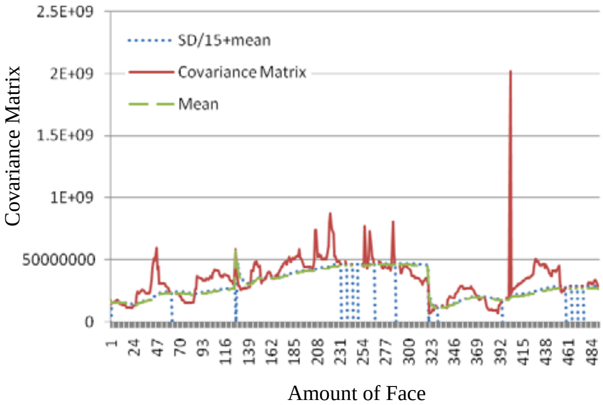
Figure 4. Best Face Quality Index

4. Experimentation and Discussion

Our system was evaluated experimentally by the following: finding m and n parameters, and testing the performance of system in the real situation. The parameters m and n play an important role to the accuracy of our best face detection method. We determine these two parameters by: firstly a learning dataset of 10 faces (male and female) with its variation (very noisy, noisy, clear, burred and very burred) are created (figure 5), and then it is used for system training in order to determine m and n parameters. We found that the optimal parameter n for the clear face image is 0 and m corresponds to 1/15. This corresponds to the first sub-window next to the mean value with the larger size equal to $1/15\,\sigma$, meaning that all face images having σ_i greater than μ_{κ_i} and less than $\mu_{\kappa_i} + \sigma_{\kappa_i}/15$ will be considered as a face having the best quality.

Figure 5. Learning dataset

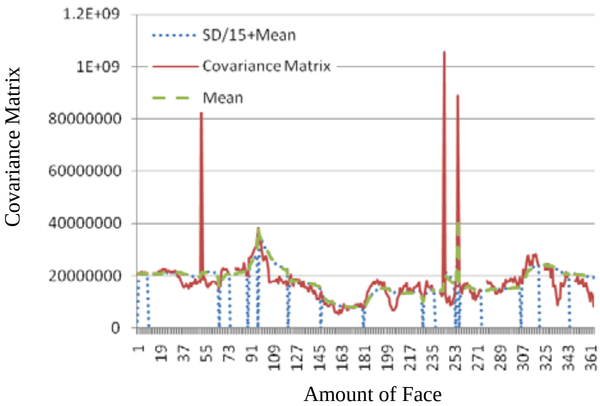
For testing the performance of our method, we created a testing dataset of four video sequences with one minute each at 30 fps. Each sequence corresponds to a moving person in an indoor environment which contains the testing characteristics: (1) burred face images from fast moving, or far distance from camera, (2) noisy images with poor light condition, and (3) clear face images when no motion with a good light condition and close to the camera. Figures 6, 7, 8 and 9 illustrate graphs of the face quality index $\kappa_i(\mathbf{I}_f)$, its mean $\eta_{\mu,\sigma^2}(\kappa_i)$, and the best quality image detected by function $h(\kappa_i)$. Visually, we can notice that the quantity of $\kappa_i(\mathbf{I}_f)$ value for each sequence is difference which relatively depends on the face skin texture and environment. So, the best quality face determined by our method doesn't mean the most clear face image of that person, but it represents the best one at that period of time. We observed experimentally that our method give quite good results of selected face. So, the much burred and very noisy images are rejected completely.



(a)

(b) (c) (d)

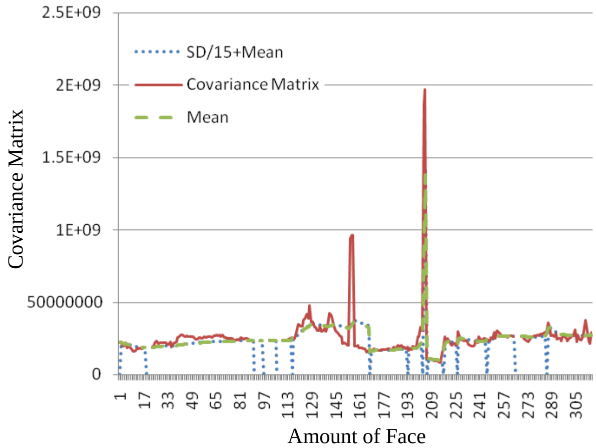
Figure 6. Sequence#1 (a) graph of $\kappa_i(\mathbf{I}_f)$, $\eta_{\mu,\sigma^2}(\kappa_i)$ and $h(\kappa_i)$; (b,c) simple frames from sequence; and (d) an example of selected face.



(a)

(b) (c) (d)

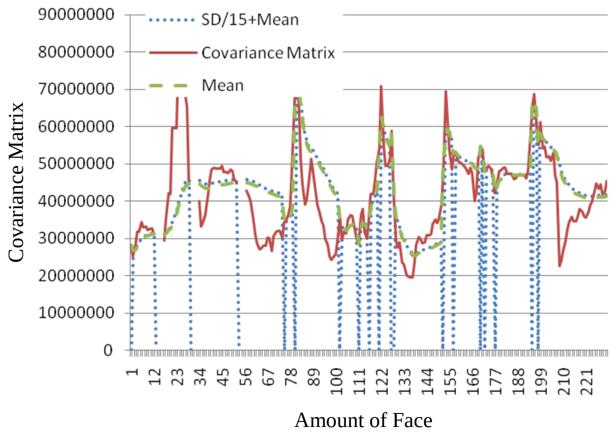
Figure 7. Sequence#2 (a) graph (b,c) simple frames from sequence; and (d) an example of selected face.



(a)

(b) (c) (d)

Figure 8. Sequence#3 (a) graph (b,c) simple frames from sequence; and (d) an example of selected face.



(a)

(b) (c) (d)

Figure 9. Sequence#4 (a) graph (b,c) simple frames from sequence; and (d) an example of selected face.

5. System Integration

Our algorithm is integrated with a video analysis framework proposed in [6][7]. This framework allows our algorithm to analysis image sequences from IP-cameras which is executed on the multi-cores and multi-processors architecture. The framework architecture is divided into five important components: video acquisition, video analysis, video encoding, event alert, and playback. These components lay on three layer architectures: Application Layer (AL), Communication Layer (CL) and Processing Layer (PL), shown in figure 10. Note that Processing Component (PC) means alls component located on PL, and respectively for Communication Component (CC). The CL facilitates the communication between components, and also with external sources such as cameras, alert system and database server. It emphasizes how images would be transferred from one to another component. All PC components are developed separately from the framework and acts as plug-ins. We implement our algorithm as a PC component of this framework.

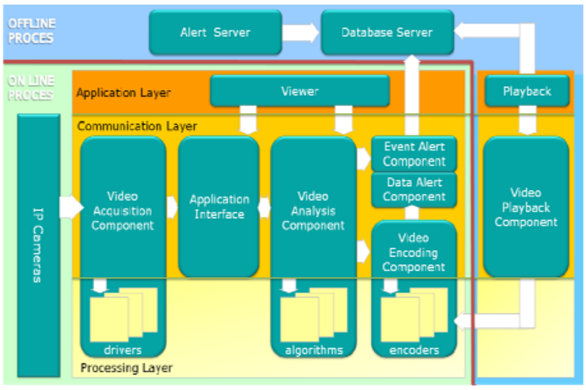


Figure 10. Video Analysis Framework using for our algorithm

The software architecture of framework is entirely implemented under the .NET framework version 2.0. It consists four layers (figure 11): application layer, application core layer, system core layer, and external module layer. The *VsCore* in the system core layer is the most important library that implements how to synchronize between communication components for supporting the three possible types of analysis: local (*VsCamera*), cooperative (*VsChannel*) and global analysis (*VsPage*). The system core also defines how external processing modules can be interfaced and executed during the runtime service. Four types of interfaces are predefined: *VsProvider* for image acquisition, *VsAnalyzer* for any kind of image filters, *VsEncoder* for image encoding into files, and *VsData* for database connection.

For the application core, its role is to define the functionality of a specific application that implements under the system core. The application layer interfaces the application core in different ways to the users which allows, the same function can be called from either standalone or web application. Figure 12 shows the standalone *VsMonitor* app interface that allows viewing analysing (using our algorithm) and encoding the image sequence from IP-cameras.

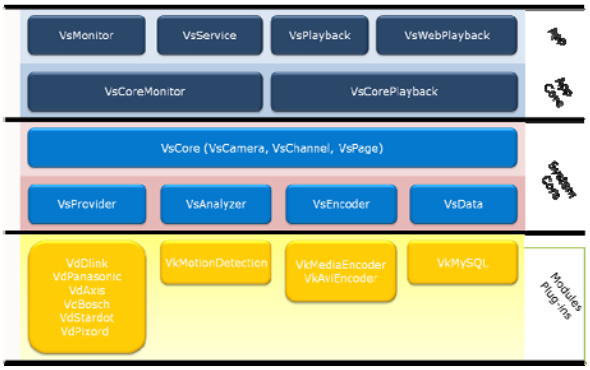


Figure 11. Software architecture

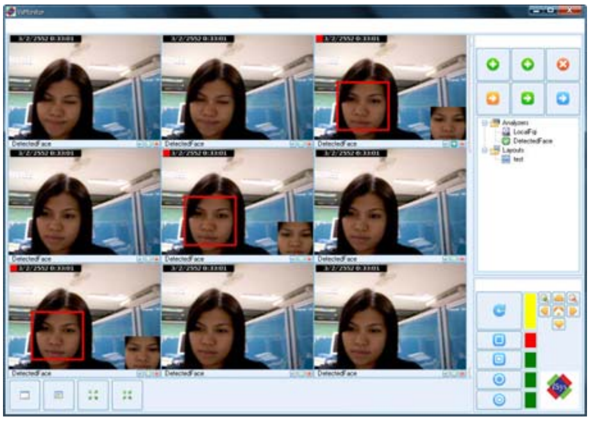


Figure 12. VsMonitor with our algorithm implemented as a PC module

6. Conclusion

We presented in this paper a method for face quality index measurement based on the covariance matrix applied for the key frame extraction. The face quality index is determined depending on face skin characteristics and its environment. The encouraging results show some advantages of our method that also could be run in real-time for the real world application.

References

[1] Ntalianis, K.S. and Kollias, S.D., An Optimized Key-Frames Extraction Scheme Based on SVD and Correlation Minimization, ICME 2005, 6-6 July 2005 Page(s):792 - 795

[2] Method of selecting key-frames from a video sequence, patent United States Patent 7184100

[3] Viola, P. and Jones, M, Rapid object detection using a boosted cascade of simple features, CVPR2001, I-511- I-518 vol.1

[4] Gary R Bradski, Computer Vision Face Tracking For Use in a Perceptual User Interface, Intel Technology Journal, No.Q2. 98

[5] N. Suvonvorn and Anant Chocksuriwong, Real-Time Face Detection/Identification for Surveillance System, IEEE International Conference on Electronic Design, Malaysia, 1-3 December 2008

[6] N. Suvonvorn, 2008, A Video Analysis Framework for Surveillance System, IEEE International Workshop on Multimedia Signal Processing, Cairns, Queensland, Australia, 8-10 October 2008.

[7] N. Suvonvorn, 2008, "A Generic Surveillance System", The Sixth PSU-Engineering Conference, Songkhla, Thailand, 8-9 May 2008