We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

186,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Chapter 7

Side View Driven Facial Video Coding

Ulrik Söderström and Haibo Li

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/53101

1. Introduction

Video compression is a task that is becoming more and more interesting as the use of video is growing rapidly. Video is today part of the daily life for anyone with a computer, e.g., video sites as Youtube [1] and video commercials on regular web sites. The network capacity is growing and one could assume that this would reduce the need for more efficient video compression. But since the use of video, and other types of data, is growing even faster than the capacity growth better ways for compression are needed. Furthermore, video is wanted for low capacity networks where high compression of video is essential for transmission so the need of efficient video compression is therefore also growing. The users are becoming used to a certain service and expects to have the same kind of service everywhere.

Video compression based on Discrete Cosine Transform (DCT) and motion estimation (ME) has almost reached its potential when it comes to compression ratio so to supply the users with the video capacity they want it is necessary to use different kinds of video coding techniques than the ones which are used today. These techniques function very well at reasonably high bitrates but when the bitrate is very low they fail to work properly. Of course, it is desirable to have a video compression scheme that works on arbitrary video but in this work we focus on facial video; video where the facial mimic is the most prominent and important information.

We have previously presented a coding scheme based on principal component analysis (PCA) [2] that make use of the fact that it is the facial mimic which is the most important part for facial video [3]. We have also calculated theoretical boundaries for the use of such a coding scheme [4]. We have extended the use of PCA into asymmetrical PCA (aPCA) [5] where a part of the frame is used for encoding and the entire frame is decoded. In this work we will show how aPCA can be used for encoding of one part of the frame while decoding uses a different part of the frame. More specifically we will use the side view or the profile of the side view of a face for encoding and decode the frontal view of this face.

INTECH
open science | open minds

© 2012 Söderström and Li; licensee InTech. This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

(a) Frontal view camera (b) Side view camera

Figure 1. The face captured by cameras filming the frontal and side view of a person.

Encoding with the side view gives a huge benefit in usability since the user can wear a camera that films the side of the face instead of the front of the face. From a user point of view this turns video capturing into something which is performed without any inconvenience such as a camera in front of the face would mean. The differences are visualized in Fig. 1. For communication purposes you want to see the frontal view of a face. Humans usually wants to talk to someone who is facing them.

Communication with video where you can see the face of the person you are communicating with has not become the important application that is was predicted to be. Video telephone calls are used much less then it was thought. But still, the use of visual communication is growing over the Internet, with the use of popular communication applications, e.g. Skype. For this kind of communication the users usually sit in front of their computers with a fixed web camera; thus enabling hands-free usage. With aPCA hands-free usage can be realized even for mobile users. For all users the bitrate for transmitting the video can be drastically lowered so the scheme is not only beneficial for mobile users; low bitrate is wanted in all usage settings.

For some tasks video is superior to only voice since you can infer a lot of information from the facial appearance. If it can be provided when the user has both hands free and can move freely, it will improve the quality of communication. If the user has to hold a camera or be positioned in front of a camera you limit the users freedom to move as they have to be positioned in front of a fixed camera. The side view of a face can easily be filmed and this view can be used to encode video while the frontal view of the face is decoded.

The video coding that we present in this article describes a new way to encode video; the decoded part is not even used for encoding. This idea has been used for several other techniques (see section 2) but in these implementations real video is not decoded; they provide an avatar, an animation or a cartoon-like frame. aPCA coding creates real video

frames (frontal view) from a video content that is not decoded (side view) but is much easier accessible for encoding.

Section 2 describes related work in very low bitrate video compression and section 3 describes PCA video coding. Section 4 explains how two views are used for encoding and decoding while section 5 show how only partial information from one view can be used for encoding while the entire second view is decoded. Section 6 show the results from practical experiments and the work is concluded in section 7.

2. Related work

Discrete Cosine Transform (DCT) is regarded as state-of-the-art for video compression and it is used together with motion estimation in most standard video codecs [6, 7]. But such a representation requires bitrates which are too high for low capacity networks, such as GSM. And as a consequence, video encoded with DCT does not have enough quality at very low bitrates. A previous technique that was aimed at videoconferencing at low bitrates based on DCT is called H.263 [8, 9]. Since this solution was based on DCT it didn't provide good enough quality when the bitrate was as low as over mobile networks. There are several other ways to represent facial images so that it can be transmitted over low capacity networks. Matching pursuit (MP) [10] is a technique that uses an alphabet where the encoder divides the video frames into features and the decoder assembles the features to a reconstructed frame. Very low bitrate is achieved by only transmitting information about which features that the frame consists of. A wireframe that resembles a face is used by several techniques, e.g., MPEG4 facial animation [11] and model based coding [12, 13]. The wireframe is texture-mapped with a facial image to give it a more natural appearance. Very low bitrate is achieved since only information about how to alter the wireframe is needed to make the face change shape between frames. The wireframe consists of several polygons; one of the most popular being the CANDIDE model, which comes in several versions [14]. Active Appearance Model (AAM) [15] relies on a statistical model of the shapes and pixel intensity of a face for video compression. AAM are statistical models; models of the shape of an object. The models are iteratively deformed so that they fit an object. For facial coding a model of the facial features are mapped onto a facial image. The model cannot vary in any possible way; it is constrained by the changes which occur within a training set. A comparative study of several AAM implementations can be found in [16]. Since the introduction of Microsoft Kinect sensor [17] which provide low-cost depth images solutions that use the Kinect to add quality to the extraction of facial features have been implemented. A better extraction of the facial features enable a much more accurate use of them for visualization purposes. An example of such an implementation is provided by Weise et al. [18] where they use facial animation as output and video recorded with the Kinect sensor as input.

The techniques which are noted above have at least one vital drawback when it comes to usage in visual communication. The face exhibits so many tiny creases and wrinkles that it is impossible to model with animations or low spatial resolution so high quality video is superior to animations [19]. To preserve the resolution and natural look of a face Wang and Cohen used teleconferencing over low bandwidth networks with a framerate of one frame each 2-3 seconds [20]. But to sacrifice framerate to preserve resolution is not acceptable either since they are both important for many visual tasks [21]. Any technique that want to provide

video at very low bitrates must be able to provide video with high spatial resolution, high framerate and have natural-looking appearance. We have previously shown that PCA can overcome these deficits for encoding of facial video sequences [3]. Both Crowley [22] and Torres [23] have also made implementation where they use PCA to encode facial images. We have further improved PCA coding with asymmetrical PCA (aPCA) [5]. In the following sections we describe video coding based on PCA and aPCA. More extensive descriptions can be found in the references.

3. Principal component analysis video coding

Video coding based on Principal Component Analysis (PCA) is described in this section. A more detailed description and examples can be found in [3].

It is possible to decompose a video sequences into principal components and represent the video as a combination of these components. There are as many possible principal components as there are video frames N and each principal component is in fact an image. The space containing the facial images is called eigenspace Φ and this will be a space for a person's facial mimic when the eigenspace is extracted from a video where a person is displaying the basic emotions. Ohba *et.al.* provides a detailed explanation of such a personal mimic space [24]. The eigenspace $\Phi=\{\phi_1\ \phi_2\ ...\ \phi_N\}$ is constructed as

$$\phi_j = \sum_i b_{ij}(\mathbf{I}_i - \mathbf{I}_{\underline{0}}) \tag{1}$$

where b_{ij} are eigenvalues from the eigenvectors of the covariance matrix $\{(\mathbf{I}_i - \mathbf{I}_{\underline{0}})^T(\mathbf{I}_j - \mathbf{I}_{\underline{0}})\}$. $\mathbf{I}_{\underline{0}}$ is the mean of all video frames and is constructed as:

$$\mathbf{I}_{\underline{0}} = \frac{1}{N}\sum_{j=1}^{N}\mathbf{I}_j \tag{2}$$

Projection coefficients $\{\alpha_j\}$ can be extracted from each video frame through projection:

$$\alpha_j = \phi_j(\mathbf{I} - \mathbf{I}_{\underline{0}})^T \tag{3}$$

It is then possible to represent a video frame as a combination of the principal components and the mean of all pixels. When all N principal components are used this representation is error-free:

$$\mathbf{I} = \mathbf{I}_{\underline{0}} + \sum_{j=1}^{N}\alpha_j\phi_j \tag{4}$$

The model is very compact and several principal components can be discarded with a very small error. A combination with fewer principal components M can be used to represent the image with a small error.

$$\hat{\mathbf{I}} = \mathbf{I}_{\underline{0}} + \sum_{j=1}^{M} \alpha_j \phi_j \tag{5}$$

where M is a selected number of principal components used for reconstruction ($M < N$).

The sender and receiver use the same model for encoding and decoding and the only thing that needs to be transmitted between them are the projection coefficients. The sender extracts coefficients with the model and the receiver uses the coefficients with the model to recreate the video frames. This is very similar to the way that DCT-coding works since the sender and receiver uses the same model and only coefficients need to be transmitted.

The extent of the error incurred by using fewer components (M) than (N) is examined in [4]. With the model it is possible to encode entire video frames to only a few coefficients $\{\alpha_j\}$ and reconstruct the frames with high quality.

4. Asymmetrical principal component analysis video coding with encoding through the side view

There are two major drawbacks with the use of full frame encoding:

1. The information in the principal components are calculated from every pixel located in the frames. Pixels that aren't important for the facial mimic or belong to the background will have a large effect on the model if they have high variance.
2. The encoding and decoding complexity as well as the complexity for calculating the principal components are directly dependent on the number of pixels used in the calculations. High spatial resolution means a high complexity.

We have previously presented asymmetrical principal component analysis (aPCA) [5] where we use one part of a frame for encoding and decode the entire frame. This is possible to achieve through the use of pseudo principal components; information where not the entire frame is a principal component.

Previously we used foreground and the entire frame for aPCA. In this work we use a different kind of video sequences; where there are two facial views in the video. We have the side view of a face $\mathbf{I}^s$ and the frontal view of the face $\mathbf{I}^{fr}$ (Fig. 2).

We use the side view $\mathbf{I}^s$ for encoding and the frontal view $\mathbf{I}^{fr}$ for decoding. This is possible when there is a correspondence between the facial features in the two views. An eigenspace for the side view is constructed as:

$$\phi_j^s = \sum_i b_{ij}^s (\mathbf{I}_i^s - \mathbf{I}_{\underline{0}}^s) \tag{6}$$

Figure 2. A video frame with the side $\mathbf{I}^s$ and front view $\mathbf{I}^{fr}$ shown.

where b_{ij}^s are eigenvalues from the eigenvectors of the covariance matrix $\{(\mathbf{I}_i^s - \mathbf{I}_{\underline{0}}^s)^T(\mathbf{I}_j^s - \mathbf{I}_{\underline{0}}^s)\}$ and $\mathbf{I}_{\underline{0}}^s$ is the mean of the side view. Encoding of video is performed as:

$$\alpha_j^s = (\mathbf{I}^s - \mathbf{I}_{\underline{0}}^s)^T \phi_j^s \tag{7}$$

where $\{\alpha_j^s\}$ are coefficients extracted using information from the side view $\mathbf{I}^s$.

Since the frontal view should be decoded instead of the side view a space for decoding is needed. This is a space consisting of pseudo principal components where no no part of the components are orthogonal.

$$\phi_j^{p_{fr}} = \sum_i b_{ij}^s (\mathbf{I}_i^{fr} - \mathbf{I}_{\underline{0}}^{fr}) \tag{8}$$

where $\mathbf{I}^{fr}$ is the frontal view of the frames and $\mathbf{I}_{\underline{0}}^{fr}$ is the mean image of the frontal view.

The coefficients from encoding with the side view are combined with this space for decoding.

$$\hat{\mathbf{I}}^{fr} = \mathbf{I}_{\underline{0}}^{fr} + \sum_{j=1}^{M} \alpha_j^s \phi_j^{p_{fr}} \tag{9}$$

So, a decoded video of the frontal view can be created based only on information from the side view. The desired information is available through a much more easily-accessed

information; this information is not even needed at the encode side. The side view is used for encoding and through the aPCA model the decoder can create the frontal view. The information that should be decoded is not needed for encoding.

aPCA models the correspondence between the views and it is easy to realize that such a correspondence exists. When, e.g., the mouth opens it gives rise to the same amount of change in the frontal and side view. The coefficients extracted from Eq. 7 are extracted from the change in the side view and it is the same change in the frontal view that is reconstructed in Eq. 9. Such reconstruction enables a easier use of hands-free equipment because it is more comfortable and easy to film the side of the face instead of the front of the face. The frontal view is always used for communication purposes so a decoded frontal view is fundamental for communication media. For communication through web cameras the sense of having eye-contact is often lost since the camera is not positioned in the screen; where the image(s) of the other(s) are shown. With a model where the eyes are looking at a screen this sense can be available since the information which is used for encoding does not affect the position of the decoded eyes.

The complexity for encoding is directly dependent on the spatial resolution of the frame that should be encoded. The important factor for complexity is $K * M$, where K is the number of pixels and M is the chosen number of eigenvectors. The complexity is reduced when two different views are used for encoding and decoding and there are fewer pixels in the view used for encoding.

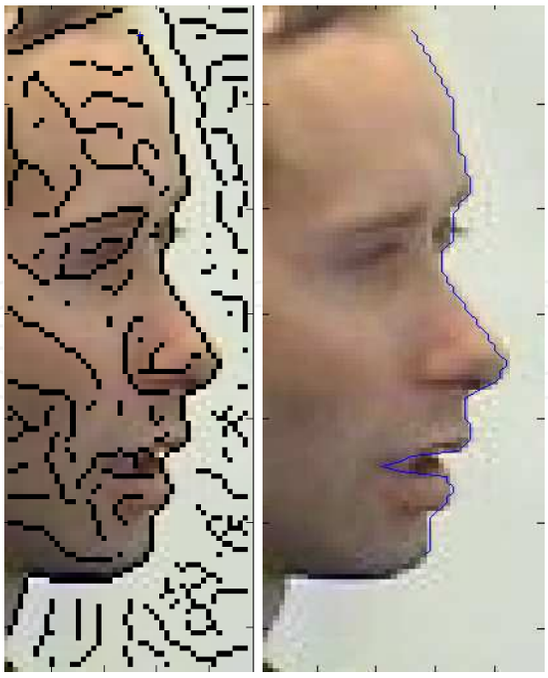
5. Asymmetrical principal component analysis video coding with encoding through the profile

Instead of using the side view $\mathbf{I}^s$ for encoding we use the profile of the side view $\mathbf{X}^{pr}$. The profile of the side view is calculated as the pixel position of the edge between the face and the background in the side view and it is extracted through edge detection.

5.1. Edge detection

The profile is extracted from the side view through edge detection. Edge points are found by applying canny edge detection [25] to the image. The canny edge detector marks several edges in the picture (Fig. 3(a)) and the ones representing the edges between the side of the face and the background are selected by examining the pixel neighbors. The first pixel of the edge is chosen and each neighboring pixel which is an edge can be the next edge pixel. Clockwise pixels are selected ahead of counter-clockwise pixels and the selected edge for the entire side view is chosen. The facial mimic is dependent on changes in the area between the forehead and the mouth so the profile is calculated in this area. The pixel positions for the lip area are manually calculated as the edge of the lip instead of the edge between the face and the background. For each vertical position we extract one horizontal position (Fig. 3(b)). The side view $\mathbf{I}^s$ consists of the pixel intensities in the image:

$$\mathbf{I}^s = \left[I(x_1, y_1)\ I(x_2, y_1)\ I(x_1, y_2)\ \ldots\ I(x_h, y_v) \right] \tag{10}$$



(a) Possible profile points (b) Full profile with mouth segmented

Figure 3. Edge detection process. (a) Possible edges. (b) Chosen edge.

where *I(x,y)* is the intensity for the specific pixel and h and v are the horizontal and vertical size of the images respectively. The profile only consist of the positions for the edges:

$$\mathbf{X}^{pr} = \left[x_{e_1}, y_{e_1}\ x_{e_2}\ x_{e_3}\ \ldots\ x_{e_T}\right] \tag{11}$$

where T is the number of points in the profile. The eigenprofile for encoding is calculated according to:

$$\phi_j^{pr} = \sum_i b_{ij}^{pr}(\mathbf{X}_i^{pr} - \mathbf{X}_{\underline{0}}^{pr}) \tag{12}$$

where $\mathbf{X}^{pr}$ are the profile of the side view, b_{ij}^{pr} are eigenvalues from the eigenvectors from the covariance matrix $\{(\mathbf{X}_i^{pr} - \mathbf{X}_{\underline{0}}^{pr})^T(\mathbf{X}_j^{pr} - \mathbf{X}_{\underline{0}}^{pr})\}$ and $\mathbf{X}_{\underline{0}}^{pr}$ is the mean image of the profile positions. $\mathbf{X}_{\underline{0}}^{pr}$ is shown in Fig. 4 and the first three eigenprofiles ϕ_j^{pr} are shown in Fig. 5.

A space for the frontal view is calculated similarly to when the profile is used for encoding as when the side view is used for encoding. The difference is that the eigenvectors from $(\mathbf{X}^{pr} - \mathbf{X}_{\underline{0}}^{pr})^T(\mathbf{X}^{pr} - \mathbf{X}_{\underline{0}}^{pr})$ are used instead of the eigenvectors from $(\mathbf{I}^s - \mathbf{I}_{\underline{0}}^s)^T(\mathbf{I}^s - \mathbf{I}_{\underline{0}}^s)$.

$$\phi_j^{p_{pr}} = \sum_i b_{ij}^{fr}(\mathbf{I}_i^{fr} - \mathbf{I}_{\underline{0}}^{fr}) \tag{13}$$

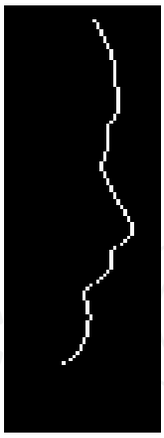
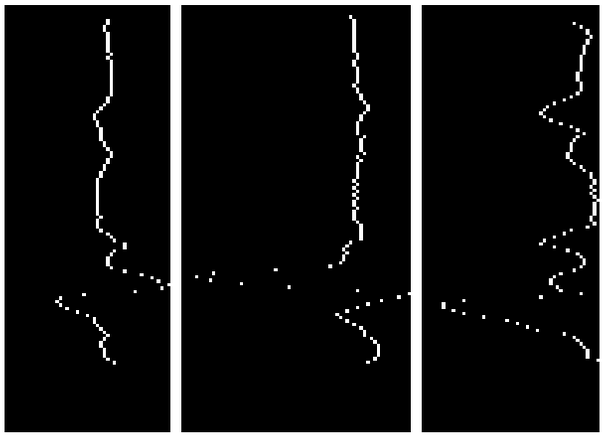


Figure 4. The mean profile $\mathbf{X}_{\underline{0}}^{pr}$.



(a) ϕ_1^{pr} (b) ϕ_2^{pr} (c) ϕ_3^{pr}

Figure 5. The first three eigenprofiles ϕ_j^{pr} (Scaled for visualization).

Encoding is then performed according to:

$$\alpha_j^{pr} = (\mathbf{X}^{pr} - \mathbf{X}_{\underline{0}}^{pr})^T \phi_j^{pr} \tag{14}$$

and decoding is performed as:

$$\hat{\mathbf{I}}^{fr} = \mathbf{I}_{\underline{0}}^{fr} + \sum_{j=1}^{M} \alpha_j^{pr} \phi_j^{p_{fr}} \tag{15}$$

In this way the profile is used for encoding and the frontal view is decoded. The profile can be extracted through low-level processes such as edge detection which makes it fast to find. The success of edge detection may depend on the background so a template matching combined with edge detection is much more stable. For the experiments in this work we have manually corrected some edge detection errors.

	Reconstruction quality (PSNR)		
ϕ	Y	U	V
5	34,9	39,3	41,9
10	37,4	39,4	42,2
15	38,6	39,4	42,3
20	39,6	39,4	42,3
25	40,3	39,4	42,3

Table 1. Reference results for frontal view encoding and decoding of the videos.

6. Practical results

In this section we present practical results for encoding with the side view and profile of the side view. The reconstructed video is always the frontal view. We use video sequences which contain both the side and frontal view of a person (Fig. 2). In these frames there is a correspondence between the facial features in the side view $\mathbf{I}^s$ and the frontal view $\mathbf{I}^{fr}$. For example, when the mouth is opened it is visible in both views and the change in the side view is consistent with the change in the frontal view. For both experiments we use 10 video sequences. Each video sequence show one person when he/she is displaying Ekman's six basic expressions. The sequences are approximately 30 seconds long and the subjects starts with a neutral expression. After each basic expression the subject returns to the neutral expression. The reported results are averages for all the sequences.

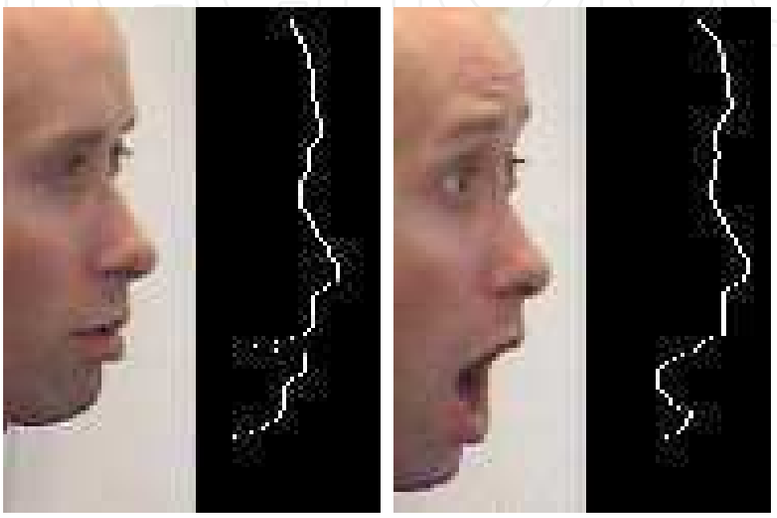
The information that should be decoded is not needed for encoding since this information can be found by looking at information that has correspondence with it. The reconstruction quality is measured for the frontal view only and not the entire frame since it is only this view which is reconstructed. The reconstruction quality for encoding with the side view and the profile are compared to the quality of the frontal view when this is also used for encoding (regular PCA video coding). The quality for using regular PCA video coding is shown in Table 1. Table 2 and 3 show the reduction in quality there is compared to Table 1. The quality is measured in YUV color space since the video sequences are coded in this color space. The complexity reduction from using the alternative views for encoding instead of the frontal view is presented for the two different experiments.

6.1. Encoding with the side view, decoding with the frontal view

Table 1 show the quality of reconstructed frontal view for regular PCA encoding and Table 2 show the reduction in quality compared to this Table when the side view is used for encoding. From Table 2 it can be seen that the quality of the reconstructed frontal view is slightly lower for side view encoding compared to frontal view encoding (1,5 dB).But at the same time the encoding complexity is reduced. With a spatial size for the frontal view $\mathbf{I}^{fr}$ of 94x144 and a side view size of 48x128 ($\mathbf{I}^s$) the complexity reduction is almost 55 %. However, the usefulness of this encoding is not found in quality or complexity since it provides a new idea for video coding. All previous methods aim at reconstructing the same video as the original video from a compressed version. Here it is possible to make use of a different video for encoding compared to decoding. This video is much easier to record and the usefulness of the video coding will be increased vastly.

	Red. of rec. qual. (PSNR)		
ϕ	Y	U	V
5	0,8	0,1	0,0
10	1,4	0,1	0,1
15	1,6	0,0	0,1
20	1,7	0,0	0,1
25	1,6	0,0	0,1

Table 2. Reduction of reconstruction quality for encoding with side view compared to encoding with frontal view.



(a) Anger (b) Fear

Figure 6. Example frames of profiles relating to the side view.

6.2. Encoding with the side view profile, decoding with the frontal view

Instead of using the side view $\mathbf{I}^s$ for encoding we use the profile of the side view $\mathbf{X}^{pr}$. This profile is calculated according to section 5. Examples of how the profiles relate to the side views are shown in Fig. 6.

The amount of data that is needed for encoding is reduced from each pixel in the side view to only the positions of the profile points. In our experiments the profile consists of 98 points (T) so the encoding information is only 99 values. X values for all positions are needed but only the first y value since you only add one to the next (25 follows 24, 26 follows 25 and so on). The complexity reduction compared to encoding with the frontal view is more than 99%.

The objective quality reduction compared to encoding with the frontal view (Table 1) is presented in Table 3. A figure of the Y channel quality for the different encoding options is shown in Fig. 7. The reconstruction quality is significantly lower when the profile is used for encoding compared to using the side view. When the video is evaluated subjectively it can be seen that the video becomes jerky and loses its natural smoothness; something that is difficult to visualize with still images. In Fig. 8 a comparison between the original and reconstructed frames is shown together with the profiles. Most frames are reconstructed quite well but some video frames are not consistent with the rest of the video and this reduces the reconstruction quality. We have previously shown how this issue can be handled

	Red. of rec. qual. (PSNR)		
ϕ	Y	U	V
5	2,5	0,1	0,1
10	3,8	0,2	0,2
15	4,2	0,2	0,2
20	4,5	0,2	0,2
25	5,0	0,2	0,3

Table 3. Reduction of reconstruction quality for encoding with profile compared to encoding with frontal view.

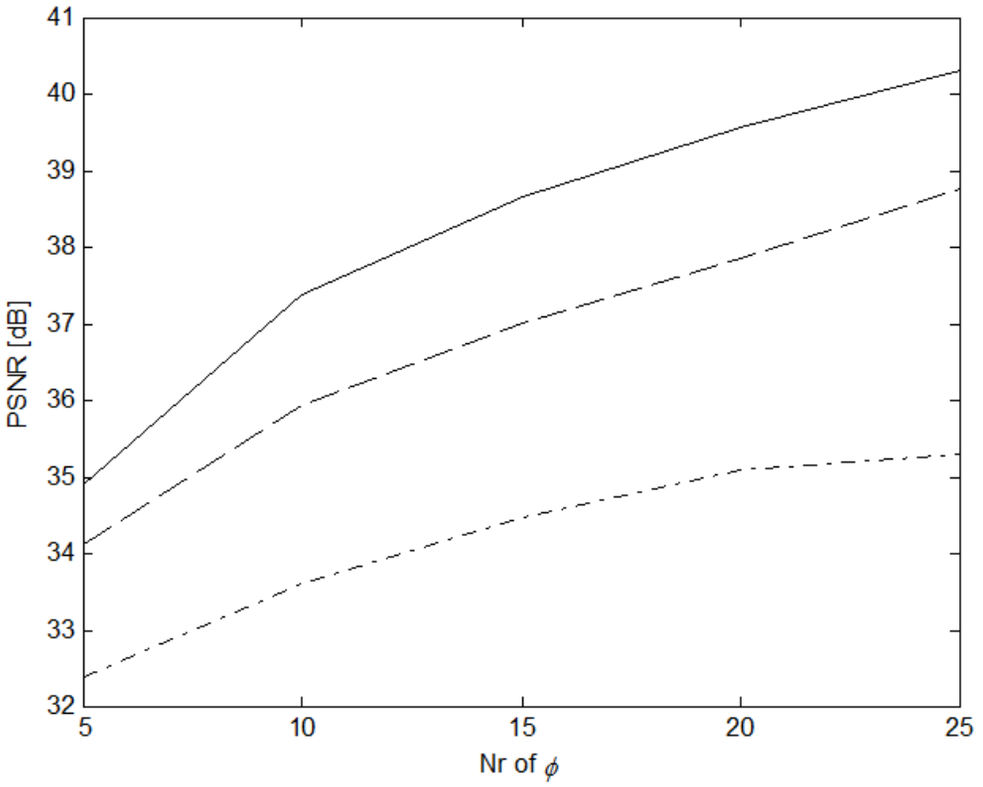


Figure 7. The Y-channel quality. – *Encoding with the frontal view* I^{fr} - - *Encoding with the side view* I^{s} -·- *Encoding with the frontal view* X^{pr}.

with Local linear embedding (LLE) [26, 27]. LLE is an unsupervised learning algorithm that computes low-dimensional, neighborhood-preserving embeddings of high-dimensional sources. By creating such an embedding for the reconstructed facial mimics it is possible to ensure that the frames share similarities with the surrounding frames and thus create a video with smooth transitions between the frames. The quality, measured in PSNR, will then be increased. Some frames are reconstructed with a blurry result and this also lowers the PSNR, but most of the quality loss depends on the difference in facial expression between the original and reconstructed frames.

http://dx.doi.org/10.5772/53101

(a) Original frames

(b) Profiles

(c) Reconstructed frames

Figure 8. Example frames of original frames, profiles and frames reconstructed from encoding with the profiles.

7. Discussion

We have shown how Asymmetrical Principal Component Analysis (aPCA) is used for encoding and decoding of different facial parts. The importance is to have a correct correspondence between facial features in different views of the face. The use of the side view for encoding will increase the usefulness of aPCA since we can use video sequences which are easier to record for encoding but still decode the frontal view of the face. The side view is easier to film since a camera can be placed closely to the side of the face instead of in front of the face.

We have furthermore shown that it possible to use the position of the profile for encoding instead of the pixel values in the side view. The reconstructed video loses its natural transition between frames and the objective quality drops. This can be solved by incorporating local linear embedding and dynamic programming. The amount of data used

for encoding is reduced with more than 99 % when the profile is used compared to using the frontal view.

Encoding from the side view and decoding of the frontal is a new kind of video coding since the decoded information is not even used in the encoding process. This enables the use of more user friendly video acquisition since the frontal view does not need to be recorded and the side view can be recorded with less user impact.

In the profile there is very little information about the eyes; it is difficult to model where the eye is looking. In the profile the left eyelid is visible so you can model opened or closed eyes but not where you are looking. As extra information to the profile it may be useful to add the pixel intensities for the eye region. This is easily solved by adding the pixels to the eigenprofile. The correspondences from both type of information are then modeled with aPCA.

Author details

Ulrik Söderström[1,⋆] and Haibo Li[2,3]

⋆ Address all correspondence to: ulrik.soderstrom@tfe.umu.se

1 Digital Media Lab, Dept. of Applied Physics and Electronics, Umeå University, Umeå, Sweden
2 School of Communication and Information Technology, Nanjing University of Posts and Telecommunications, Nanjing, China
3 School of Computer Science and Communication, Royal Institution of Technology (KTH), Stockholm, Sweden

References

[1] Youtube. http://www.youtube.com/ (accessed 5 May 2012).

[2] Jolliffe I. Principal Component Analysis. New York: Springer-Verlag; 1986.

[3] Söderström U, Li H. Full-frame video coding for facial video sequences based on principal component analysis. In: Proceedings of Irish Machine Vision and Image Processing Conference (IMVIP); 2005. p. 25–32.

[4] Söderström U, Li H. Representation bound for human facial mimic with the aid of principal component analysis. International Journal of Image and Graphics. 2011;10(3):343–363.

[5] Söderström U, Li H. Asymmetrical principal component analysis for video coding. Electronics letters. 2008 February;44(4):276–277.

[6] Schäfer R, Wiegand T, Schwarz H. The emerging H.264 AVC standard. EBU Technical Review. 2003;293.

[7] Wiegand T, Sullivan GJ, Bjontegaard G, Luthra A. Overview of the H.264/AVC video coding standard. IEEE Trans Circuits Syst Video Technol,. 2003;13(7):560–576.

[8] Rijkse K. H.263: video coding for low-bit-rate communication. Communications Magazine, IEEE. 1996;34:42–45.

[9] Côté G, Erol B, Gallant M, Kossentini F. H.263+: Video coding at low bit rates. IEEE Transactions on Circuits and Systems for Video Technology. 1998;8:849–866.

[10] Neff R, Zakhor A. Very Low Bit-Rate Video Coding Based on Matching Pursuits. IEEE Transactions on Circuits and Systems for Video Technology. 1997;7(1):158–171.

[11] Ostermann J. Animation of Synthetic Faces in MPEG-4. In: Proc. of Computer Animation, IEEE Computer Society; 1998. p. 49–55.

[12] Aizawa K, Huang TS. Model-based image coding: Advanced video coding techniques for very low bit-rate applications. Proc of the IEEE. 1995;83(2):259–271.

[13] Forchheimer R, Fahlander O, Kronander T. Low bit-rate coding through animation. In: In Proc. International Picture Coding Symposium PCSŚ83; 1983. p. 113–114.

[14] Ahlberg J. CANDIDE-3 - An Updated Parameterised Face. Image Coding Group, Dept. of Electrical Engineering, Linköping University; 2001.

[15] Cootes T, Edwards G, Taylor C. Active appearance models. In: In Proc. European Conference on Computer Vision. (ECCV). vol. 2; 1998. p. 484–498.

[16] Cootes T, Edwards G, Taylor C. A Comparative Evaluation of Active Appearance Model Algorithms. In: 9th British Machine Vison Conference; 1998. p. 680–689.

[17] http://www.xbox.com/en-GB/kinect (accessed 15 June 2012).

[18] Weise T, Bouaziz S, Li H, Pauly M. Realtime Performance-Based Facial Animation. ACM Transactions on Graphics (Proceedings SIGGRAPH 2011). 2011 July;30(4).

[19] Pighin F, Hecker J, Lishchinski D, Szeliski R, Salesin DH. Synthesizing Realistic Facial Expression from Photographs. In: SIGGRAPH Proceedings; 1998. p. 75–84.

[20] Wang J, Cohen MF. Very Low Frame-Rate Video Streaming For Face-to-Face Teleconference. In: DCC '05: Proceedings of the Data Compression Conference; 2005. p. 309–318.

[21] Lee J, Eleftheriadis A. Spatio-temporal model-assisted compatible coding for low and very low bitrate video telephony. In: Proceedings, 3rd IEEE International Conference on Image Processing (ICIP 96). Lausanne, Switzerland; 1996. p. II.429–II.432.

[22] Schwerdt K, Crowley J. Robust face tracking using color. In: Proc. 4th Int. Conf. on Automatic Face and Gesture Recognition; 2000. p. 90–95.

[23] Torres L, Prado D. High compression of faces in video sequences for multimedia applications. In: Proceedings. ICME '02. vol. 1. Lausanne, Switzerland; 2002. p. 481–484.

[24] Ohba K, Clary G, Tsukada T, Kotoku T, Tanie K. Facial expression communication with FES. In: International conference on Pattern Recognition; 1998. p. 1378–1378.

[25] Canny J. A Computational Approach to Edge Detection. IEEE Trans Pattern Anal Mach Intell. 1986 Jun;8(6):679–698.

[26] Le HS, Söderström U, Li H. Ultra low bit-rate video communication, video coding = facial recognition. In: Proc. of 25th Picture Coding Symposium (PCS); 2006.

[27] Chang Y, Hu C, Turk M. Manifold of Facial Expression. Analysis and Modeling of Faces and Gestures, IEEE International Workshop on. 2003;0:28.

IntechOpen

IntechOpen