Parametric Video Compression using Appearance Space

Santanu Chaudhury Subarna Tripathi Sumantra Dutta Roy
schaudhury@gmail.com subarna.tripathi@gmail.com sumantra@cse.iitd.ac.in
Electrical Engineering Department, IIT Delhi

Abstract

The novelty of the approach presented in this paper is the unique object-based video coding framework for videos obtained from a static camera. As opposed to most existing methods, the proposed method does not require explicit 2D or 3D models of objects and hence is general enough to satisfy the need for varying types of objects in the scene. The proposed system detects and tracks an object in the scene by learning the appearance model of each object online using non-traditional uniform norm based subspace. At the same time the object is coded using the projection coefficients to the orthonormal basis of the subspace learnt. The tracker incorporates a predictive framework based upon Kalman filter for predicting the five motion parameters. The proposed method shows substantially better compression than MPEG2 based coding with almost no additional complexity.

1. Introduction

Video compression is commonly achieved by removing redundancies in the frequency, spatial and temporal domains. Standard coding techniques, such as predictive coding, transform coding and vector quantization, treat the image/video as random signals and exploit their stochastic properties to achieve compression.

While most of the existing coding methods belong to the above mentioned category, there is a major research effort underway to produce video compression systems that use 2D or 3D models of objects and/or scenes and to parameterize them for compression. The prior knowledge about the domain can be used to construct models and to detect and segment objects from the scene only with the knowledge of the problem domain. One of the basic advantages of object-based video coding is that it allows the assignment of different priorities to different objects and the background in terms of coding bit-rate based on their significance in the domain of interest.

In this paper, we present a video coding frame work for videos that are acquired by a static camera. The proposed system currently extracts an arbitrary shaped object in the video frames inspired by [1]. It does not require explicit models of these objects. Instead, the system learns the model online for the object using a simple but efficient uniform subspace analysis in a moving average fashion. The basic assumption here is that the appearance space (that includes non-rigid motion, different poses, and views of the object) of an object can be represented by a small number of basis vectors such an assumption is used in literature and also justified by the experimental results presented in Section 3. Compared to other appearance based object coding framework described in [2] our method is faster and is having significantly reduced computational complexity. The incremental subspace learning in [2] is totally independent of the scheme used for tracking the object. In contrast, we propose a unified approach for tracking and appearance based object coding. In other words, the same subspace used for tracking is reused for the object coding and thus substantially reduces computation time.

The organization of the paper is as follows: In the next section, we present the object-based compression framework along with the automatic initialization for the single object to be tracked, where we discuss the encoding and decoding processes. Finally, the results and discussion of our experiments are demonstrated in Section 3.

2. Our approach

Tracking is bootstrapped by automatic detection of the moving object. Our coding solution currently can detect the moving object automatically by analyzing the first three frames, i.e. with the overhead of additional two frames buffering at the beginning of the encoding process which is quite acceptable. We have

used a moving object segmentation method based on the improved PCA which is a simplified version of the methodology used in [1]. For this technique to work the background should be still or changing slowly such as grassplot or cloud. Firstly, the principle component analysis is improved to adapt to the motion detection basically to avoid the superposition of the object in the three frames as well as to avoid the overhead of calculating the mean. The definition of traditional covariance matrix is modified to :

$$C = (X1 - X2)^T(X1 - X2) + (X2 - X3)^T(X2 - X3) + (X1 - X3)^T(X1 - X3) \quad (1)$$

Where, Xi is a one dimensional vector obtained by vectorizing the frame(i) in the sequence. Secondly, the calculation result is improved in the following way. Denote O1 and O2 as the first two eigenvectors calculated. The product of these two eigenvectors is: $O = O1 \times O2$. O effectively eliminates the superposition of the moving object in the before and after frame. And after formation of O a simple thresholding usually gives a good initialization of the object's bounding box.

After segmenting the object in the first frame of the image sequences, the object tracking is carried out by a simple way inspired by the technique described in [3] but more efficient than [3] by incorporating the predictive framework in terms of Kalman filter for the prediction of the five motion parameters.

Tracker used [5] in other object coding framework [2] is based on contour-based method which is far from real time implementation, where as the tracker [3] used in our approach runs on real-time and since tracking and coding is unified in our approach, our proposed object coding method can run on real time as opposed to [2].

At each frame, the tracker maintains an up-to-date appearance model, and the tracking task becomes a detection problem. Within the subspace framework, updating the model becomes how to define a subspace L that best *approximates* a given set of data $\{x1, \cdots, xN\}$, the observations from the previous N frames and tracking is performed by shifting the N frames window along the image sequence and repeating the tracking step in each window.

The idea behind the subspace construction for this appearance based tracking is that the uniform $L2$ reconstruction error norm

$$Error^{\infty}(L,\{x1, \cdots, xN\}) = \max_i d^2(L, xi) \quad (2)$$

is a good metric in defining linear approximations. Based on this uniform error norm, a simple algorithm is designed to update the linear subspace. For a given N observations from the previous frames, $\{x1, \cdots, xN\}$ and a positive integer k, a $D = N/k$ dimensional vector space is formed by breaking the sequence of N images into D batches of size k. For each batch i, we compute its mean mi and the subspace L is defined as the subspace spanned by these batch means, $\{m1, \cdots ,mD\}$. These vectors are not orthonormal, and in order to compute the distance between L and xi, we need to have an orthonormal basis of L. Therefore, the computationally nontrivial part of the update algorithm becomes updating the orthonormal basis. The tracking is carried out for the three color planes R, G, B in our case. The motion parameters for the non-rigid objects are predicted using Kalman filter by assuming the state dynamics to be random walk model as five motion parameters assumed to be independent of each other. Thus our simple but superior combined tracking and coding algorithm becomes as described in the following pseucode:

Tracking Algorithm: Input Parameter (Ω, N, k, S)

$\Omega = \{\omega x, \omega y, \omega w, \omega h, \omega\theta\}$ is the set of five parameters for sampling windows on the frame and S is the number of windows sampled for each frame. N is the number of previous frames retained by the tracker and k is the batch size.

Output: t, current state of the tracked object.

Internal Variable: $D = N/k$ is the (maximal) dimension of the subspace L. U is an orthonormal basis of L, m a local mean for R, G, and B space. M is the set of batch means, and it is set to be empty at beginning. $t = (x, y, w, h, \theta)$ the location of the target, which is represented by a rectangular box on the image at location (x, y) and of size (w, h) with orientation θ.

Initialization: The tracker is initialized by the bootstrapping method described. Let $x1$ be the observation at the first frame.

1. Initialize Kalman filter parameters

Tracking and coding

2. Predict t using Kalman filter: leads to speed up
3. Sample Windows: Draw S samples of windows $\{W1, \cdots ,WS\}$at various location of different orientations and sizes according to a Gaussian distribution centered at predicted t with diagonal variance specified by Ω.
4. Tracking: Rectified each window Wi to a 21-by-21 image and rasterize it to form a vector xi in R^{441}.
5. Compute the $L2$ distance between each xi and the local mean m. Choose half of $\{xi\}$ which has smaller $L2$ distance to m. Among these $S/2$ vectors, evaluate their distance to the subspace L using the orthonormal basis U for the RGB space. This rejects half of the samples and therefore, increases the speed of the tracker.

6. Let *xi* be the vector in the previous step that gives the minimal distance to *L*. The corresponding window *Wi* is then the estimate of the target for the current frame. *t* is set to *Wi*.
7. Projection coefficient of t to the learnt subspace is determined, along with the difference of the predicted and actual t and update Kalman filter parameters.
8. For each frame, the residual of t and projection coefficient are sent as the object into the stream along with the object difference image (scaled).
9. For each frame, background difference image is sent. If the background difference image is less than a threshold, then it is not sent
10. **Subspace Update**: For the interval of *k* frames, collect the observations *{x*1, · · · , *xk}* from the *k* previous frames and apply the Subspace Update Algorithm, i.e. Gram-Smith orthonormalization of the batch means of each size k of the sliding window of length N. This updates both *M* and *U*.

Figure 1: Combined Tracking and Coding Framework

For each frame, the encoder codes the estimated object $O^{t'}$ by projection coefficients onto the learned subspace and motion parameters. The background error is not transmitted in case the sum of squared differences is below a certain threshold T_b. The coded video stream, at each time instance, contains the compressed background difference image B_d, compressed object difference image O_d, and the motion parameters and appearance parameters in the R,G,B subspace of the object. It is to be noted that the appearance parameters only contain the projection coefficients of the object, rather than the basis vectors of the subspace, since the basis vectors of the subspace can be estimated using the reconstructed objects during decoding at the receiver end.

Decoding of the video is achieved by decompressing the first frame and creating a background *Bc and o*bject's appearance model is initialized. For the subsequent frames, we obtain the estimated objects $O^{t'}$ by its appearance and motion parameters, and remove the object estimation error by adding the object difference image O_d to obtain O^{t}. Furthermore, the error in the background *Bc* is removed by adding the base difference image B_d. The video frame is reconstructed by inserting the object O^t into the corrected background. Finally, the object's appearance model is updated by the same way, and the above process is repeated for all the frames. The object-based compression framework is summarized in figure 2. Philosophically almost the same decoding scheme was used in [2], but the major difference with our approach is that they have incrementally learned the appearance of the "tracked" object, i.e. they have used features for object tracking and different subspace for its coding whereas our method tracks object by the online learnt subspace which is also used for coding framework. And thus our approach leads to indeed a faster encoder.

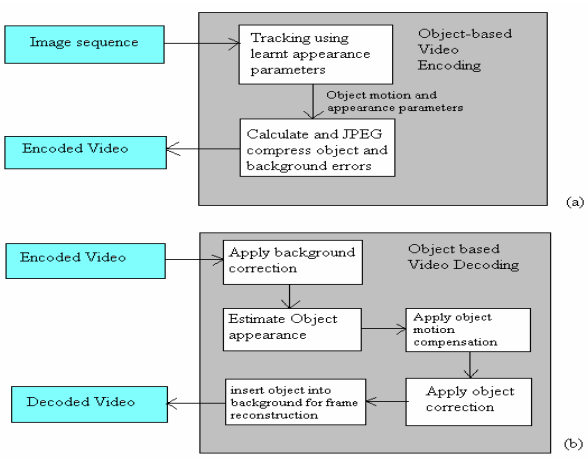

Figure 2: object-based video compression framework. (a) Encoding process. (b) Decoding process.

3. Results

Our proposed method is applied to different videos having primarily single moving object captured from a static camera (sequence length 30 to 500 frames)..

Seq	Orig Size (MB)	MP2 Size (MB)	Size In our sol	MP2 Obj psnr (dB)	Obj Psnr Our sol
hall	1.1	0.282	0.136	97.722	98.122
akio	43.5	7.33	2.49	96.957	96.657
claire	17.9	2.04	1.35	98.182	97.897
miss	5.43	1.48	1.103	97.933	97.834

Table 1: compression comparison

The constraint of single object comes only due to the automatic tracker initialization, though the concept can be easily extendible to multiple objects. MPEG2 [4] encoding is done using reference software with no rate controller (quantization parameter is set to 1 for all frames in MPEG2 as well as in our object coding framework for keeping same quality for fare comparison), and the GOP structure is (12, 3).

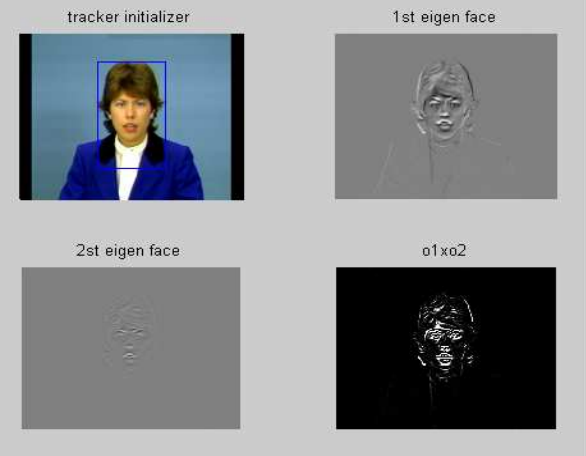

Figure 3: automatic tracker initializer for Claire_qcif

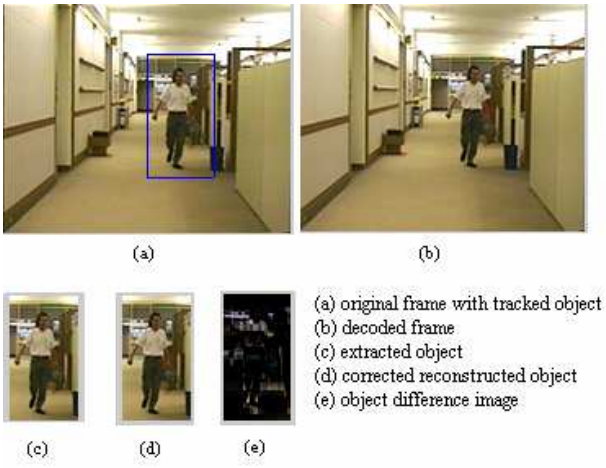

Figure 4: Results for sequence hall_qcif

Table 1 shows the compression performance of our proposed object coding framework with that of traditional MPEG2 based coding for the same output quality. Here, we assume that the Y-PSNR of the object in mpeg2 coding is same as the Y-PSNR of the whole frame. Fig 3 and fig 4 show the results of automatic bootstrapping of the tracker and object coding respectively. In cases where object size is much larger than the dimension (21x21 = 441) of the basis vectors, the residual becomes comparatively larger due to upsizing approximation, which is visible in case of Akio_cif (fig 5). The reason for not keeping the dimension of the basis vector not more than 21x21 i.e. 441 is only due to computational speed up.

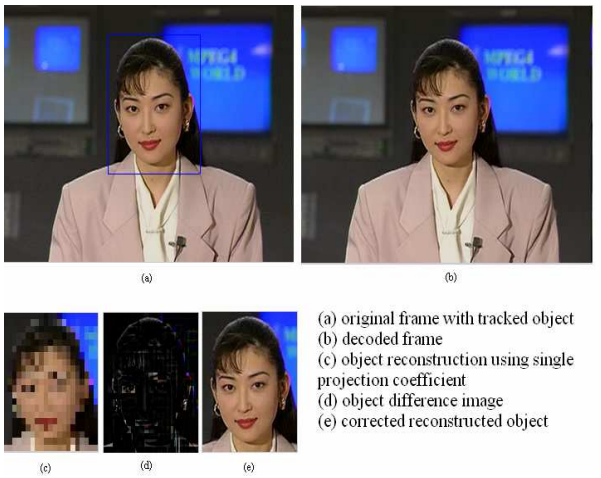

Figure 5: results for sequence Akio_cif

References

[1] Chun-Ming Li, Yu-Shan Li, Qing-De Zhuang, Qiu-Ming Li, Rui-Hong Wu, Yang Li . "Moving Object egmentation and Tracking In Video". Proc of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, 18-21 August 2005

[2] Asaad Hakeem, Khurram Shafique, and Mubarak Shah. "An Object-based Video Coding Framework for Video", *MM'05*, November 6–11, 2005, Singapore. Pages 608-617

[3] Jeffrey Ho, Kuang-Chih Lee, Ming-Hsuan Yang, David Kriegman. "Visual Tracking Using Learned Linear Subspaces". Proc of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04)

[4] ISO/IEC 13818-2: MPEG video standard, ITU-T H.262 Recommendation, 1995.

[5] Alper Yilmaz, Mubarak Shah "Contour-Based Object Tracking with Occlusion Handling in Video Acquired Using Mobile Cameras". IEEE Transactions on Pattern analysis and Machine Intelligence, Vol 26, No. 11, November 2004, pages 1531-1536