Scalable Video Compression by Employing TEMPO-SPA Arrangement along with Combined ADCT, Retaining-RLE Method

Raghu K
M Tech, Signal Processing
REVA Institute of Technology

ABSTRACT

Motion based prediction used for video coding is an efficient method in the field of video compression. But the complexity and computation time involved in this method is a burden to apply them in real time applications. In this paper, an arrangement of video frames in temporal-spatial (TEMPO-SPA) domain, which is 3D to 2D mapping of video signals is proposed. As video signals are more redundant in temporal domain compared to spatial domain, the video frames are arranged in such a manner to exploit both temporal and spatial redundancies to achieve good compression ratio. In order to reduce the time and complexity of DCT computation, Approximated DCT (ADCT) is used along with combined Retaining-RLE method. ADCT is an approximation of DCT, whose transformation matrix contains most of them zeros which reduces the number of multiplications involved in the normal DCT computation. The quantized 8x8 blocks are then encoded by combination of Retaining and Run Length Encoding (RLE) methods. Out of 64 quantized coefficients in an 8x8 block, only certain number of coefficients is retained while zig-zag scanning order and RLE is applied to this retained sequence of coefficients to reduce the data in retained sequence. Thus providing high level of compression compared to previous compression standards.

General Terms

Scalable video coding, Video compression algorithms, Efficient video signal storage and transmission, Data compression, Digital Video processing.

Keywords

Approximate DCT, Low complexity video compression, TEMPO-SPA arrangement, Retaining-RLE compression, Video compression with higher compression ratio.

1. INTRODUCTION

Nowadays in our digital world images and videos play a very important role in multimedia applications, but they contain huge amount of data. For example an image of resolution 1024x1024 requires 3145728 bytes of memory space to be stored. If a single image is very big to occupy in a memory device, then the videos which is a collection of such still image frames requires much more large memory space of 50GB range to store only a small movie clip! Hence compression of this huge amount of data in videos is very much essential. Compression is the process of compacting data into a smaller number of bits. Video compression (video coding) is the process of compacting or condensing a digital video sequence into a smaller number of bits. Compression involves a complementary pair of systems, a compressor (encoder) and a de-compressor (decoder). The encoder converts the source data into a compressed form (occupying a reduced number of bits) prior to transmission or storage and the decoder converts the compressed form back into a representation of the original video data. In block transform coding, a video is first divided into number of frames, each frame is considered as a still image and is compressed using a reversible, linear transform (such as Fourier transform). The entire video signal is converted into sequence of frames; each frame (image) is divided into non-overlapping blocks of equal size (8x8) and processing of these small blocks independently using 2-D transform is done. Linear transformations are used to map each block into a set of transform coefficients, which are then quantized and coded. For portable digital video applications, highly-integrated real-time video compression and decompression solutions are more and more required. Actually, motion estimation based encoders are the most widely used in video compression. Such encoder exploits inter frame correlation to provide more efficient compression. However, Motion estimation process is computationally intensive; its real time implementation is difficult and costly [3][4]. This is why motion-based video coding standard MPEG[11] was primarily developed for stored video applications, where the encoding process is typically carried out off-line on powerful computers. So it is less appropriate to be implemented as a real-time compression process for a portable recording or communication device (video surveillance camera and fully digital video cameras). In these applications, efficient low cost/complexity implementation is the most critical issue [4]. There are three types of data redundancies in video signal which can be eliminated to reduce the size of the video. The first type is Spatial Redundancy, which represents correlation between pixels within an image frame. This large amount of redundancy (high correlation) in an image frame is removed and can save a lot of data in representing the frame thus achieving compression. The second type is temporal redundancy, which represents correlation between pixels in successive frames in a temporal video sequence. Removing large amount of this redundancy leads to great deal of compression. Thus, researches turned towards the design of new coders more adapted to new video applications requirements. This led some researchers to look for the exploitation of 3D transforms in order to exploit temporal redundancy as well as spatial redundancy [5, 12].

2. DEFINITIONS

2.1 Two dimensional Discrete Cosine Transform (2D-DCT)

The Discrete Cosine Transform (DCT) is a time domain to frequency domain transformation. It has high energy packing efficiency close to that of the optimal Karhunen-Loeve transform. In addition, it is signal independent and can be computed efficiently by fast algorithms. For these reasons, the DCT is widely used in image and video compression.

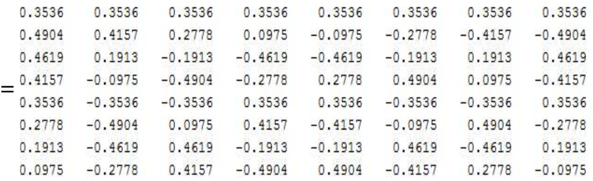
For image and video processing applications we use 2D-DCT. The forward DCT of an N x M image is calculated using the equation (1).

$$C(u,v) = \alpha(u)\alpha(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cos\left[\frac{\pi(2x+1)u}{2N}\right]\cos\left[\frac{\pi(2y+1)v}{2N}\right] \quad \ldots\ldots\ (1)$$

$$f(x,y) = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1} \alpha(u)\alpha(v)C(x,y)\cos\left[\frac{\pi(2x+1)u}{2N}\right]\cos\left[\frac{\pi(2y+1)v}{2N}\right] \quad \ldots\ldots\ (2)$$

$$\alpha(u) = \begin{cases}\sqrt{\frac{1}{N}} & for\ u = 0\\ \sqrt{\frac{2}{N}} & for\ u \neq 0\end{cases} \quad \ldots\ldots.\ (3)$$

Equation (2) represents 2D-IDCT with f(x, y) as pixel value at co-ordinate (x, y) and c(u, v) is the transform coefficient at (u, v). The Transform matrix (**A**) of an 8x8 2D-DCT is given in equation (4)



$$\mathbf{A}= \begin{matrix}
0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536\\
0.4904 & 0.4157 & 0.2778 & 0.0975 & -0.0975 & -0.2778 & -0.4157 & -0.4904\\
0.4619 & 0.1913 & -0.1913 & -0.4619 & -0.4619 & -0.1913 & 0.1913 & 0.4619\\
0.4157 & -0.0975 & -0.4904 & -0.2778 & 0.2778 & 0.4904 & 0.0975 & -0.4157\\
0.3536 & -0.3536 & -0.3536 & 0.3536 & 0.3536 & -0.3536 & -0.3536 & 0.3536\\
0.2778 & -0.4904 & 0.0975 & 0.4157 & -0.4157 & -0.0975 & 0.4904 & -0.2778\\
0.1913 & -0.4619 & 0.4619 & -0.1913 & -0.1913 & 0.4619 & -0.4619 & 0.1913\\
0.0975 & -0.2778 & 0.4157 & -0.4904 & 0.4904 & -0.4157 & 0.2778 & -0.0975
\end{matrix} \quad \ldots\ldots\ (4)$$

If **U** is an 8x8 image block, then the transformed matrix **V** is given by equation (5) and computing image block **U** from the transformed block **V** is done using equation (6).

$$\boldsymbol{V = AUA^T} \quad \ldots\ldots.\ (5)$$

$$\boldsymbol{U = A^TVA} \quad \ldots\ldots.\ (6)$$

2.2 Two dimensional Approximated Discrete Cosine Transform (2D-ADCT)

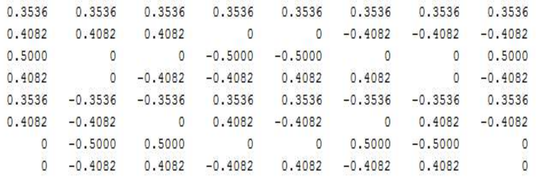
The proposed approximation method given by Renato J. Cintra and Fabio M. Bayer [1] modifies the standard DCT matrix by means of rounding off operation. Initially matrix is scaled by two and then submitted to an element wise round off operation as implemented in MATLAB. The resulting matrix **Co** has some attractive computational properties: i) its constituent elements are 0, 1 or -1, which is the indication of null multiplicative complexity. ii) As a transformation, it requires only addition being bit shift operations are absent and iii) Its scaled transpose can perform an approximate inversion. In fact, a coarse approximation matrix to outperform the DCT in a wide range of compression ratios. However the suggested approximations has some drawbacks: i) it lacks orthogonality. ii) its resulting approximation is poor, when compared with some existing methods. To overcome the above mentioned disadvantages the adjustment matrix **S** is introduced. By matrix polar decomposition theory orthogonalization of a matrix $\mathbf{C_0}$ is done with the adjustment matrix **S** given in equation (7).

$$\boldsymbol{S} = \left(\sqrt{\boldsymbol{C_0 C_0^T}}\right)^{-1} \quad \ldots\ldots\ldots\ (7)$$

The orthogonal approximated 8x8 DCT matrix (**T**) is now given by equation (8).

$$\boldsymbol{T = SC_0} \quad \ldots\ldots..\ (8)$$

The resulting 8x8 ADCT transform matrix **T** is shown below:



$$\mathrm{T}= \begin{matrix}
0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536\\
0.4082 & 0.4082 & 0.4082 & 0 & 0 & -0.4082 & -0.4082 & -0.4082\\
0.5000 & 0 & 0 & -0.5000 & -0.5000 & 0 & 0 & 0.5000\\
0.4082 & 0 & -0.4082 & -0.4082 & 0.4082 & 0.4082 & 0 & -0.4082\\
0.3536 & -0.3536 & -0.3536 & 0.3536 & 0.3536 & -0.3536 & -0.3536 & 0.3536\\
0.4082 & -0.4082 & 0 & 0.4082 & -0.4082 & 0 & 0.4082 & -0.4082\\
0 & -0.5000 & 0.5000 & 0 & 0 & 0.5000 & -0.5000 & 0\\
0 & -0.4082 & 0.4082 & -0.4082 & 0.4082 & -0.4082 & 0.4082 & 0
\end{matrix}$$

The zero value present in the **T** matrix reduces the multiplication complexity of normal DCT. Now the ADCT transformed matrix (**V**) is calculated as given in equation (9) and it's inverse to calculate 8x8 image block (**U**) from the 8x8 transform block is given in equation (10).

$$\boldsymbol{V = TUT^T} \quad \ldots\ldots.\ (9)$$

$$\boldsymbol{U = T^TVT} \quad \ldots\ldots.\ (10)$$

3. THE PROPOSED WORK

A Video signal is a sequence of still frames which are highly correlated in temporal domain i.e, in most of the video signals; the background is same in almost all the frames and only a little change in the motion of the objects exist. If we exploit this redundancy present in temporal domain in any compression algorithms, a good compression can be achieved. In JPEG technique, they split up the video into its individual frames and encode each of these frames separately. In this paper, we introduce a suitable arrangement of video frames such that the resulting representation increases the correlation among the pixels in an 8x8 block. This increased correlation boosts the compression ratio when applied with ADCT and encoded. We know that the explained temporal correlation will be more compared to spatial correlation in any video signal, therefore using temporal decomposition of video signals (instead of spatial decomposition as in the case of JPEG) achieves surprising improvement in the compression results. The different possible decompositions of a video signal are shown in figure (1).

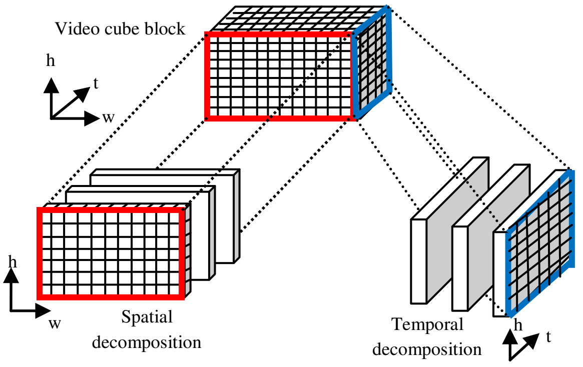


Fig 1: Spatial and Temporal decompositions of a video cube

3.1 TEMPO-SPA arrangement

The Temporal-Spatial arrangement is formed by collecting the columns of same rank from each of the 8 frames i.e, group 8 frames of a video signal into a group of pictures (GOP) and collect the first column pixels of all the 8 frames in the forward direction (1st frame to 8th frame) and group it together. Next collect the 2nd column pixels from each of these 8 frames but in the reverse direction (from 8th frame to

1st frame). This reverse direction is considered while grouping the even ranked columns because to include the spatial redundancy of the frames. Thus the arrangement proposed exploits both temporal and spatial redundancies. The following example clearly projects the TEMPO-SPA arrangement.

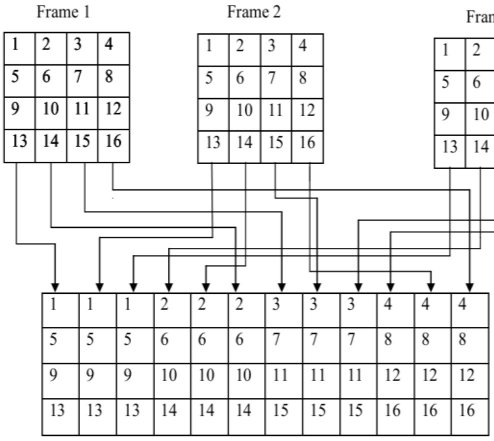


Fig 2: Example of TEMPO-SPA arrangement of 3 frames

3.2 Combined Retaining-RLE Technique

The TEMPO-SPA arranged frame are now divided into 8x8 pixel blocks, each block is transformed into frequency domain by 2D-ADCT and quantized using standard JPEG quantization table with a specified quality factor (QF). These quantized 8x8 blocks are now converted into a sequence using Zig-Zag scanning resulting in 64 quantized coefficients per each 8x8 pixel block. According to the standard zigzag sequence, only the ‘r’ initial DC coefficients were retained, with the remaining ones set to zero. If RLE is applied to this retained sequence with ‘r’ coefficients, then a sequence with length less than ‘r’ is obtained. We adopted $1 \leq r \leq 45$; if $r \leq 10$, do not perform RLE to the obtained sequence as it may increase the data instead of reducing and if $r > 10$ apply RLE to reduce the amount of data to further extent in the retained sequence. For large retaining factor chosen, the retained sequence of length ‘r’ contains many repeated quantized values, applying run-length encoding to this obviously results in still more compressed data. The overall block diagram of the proposed method is shown in figure (3).

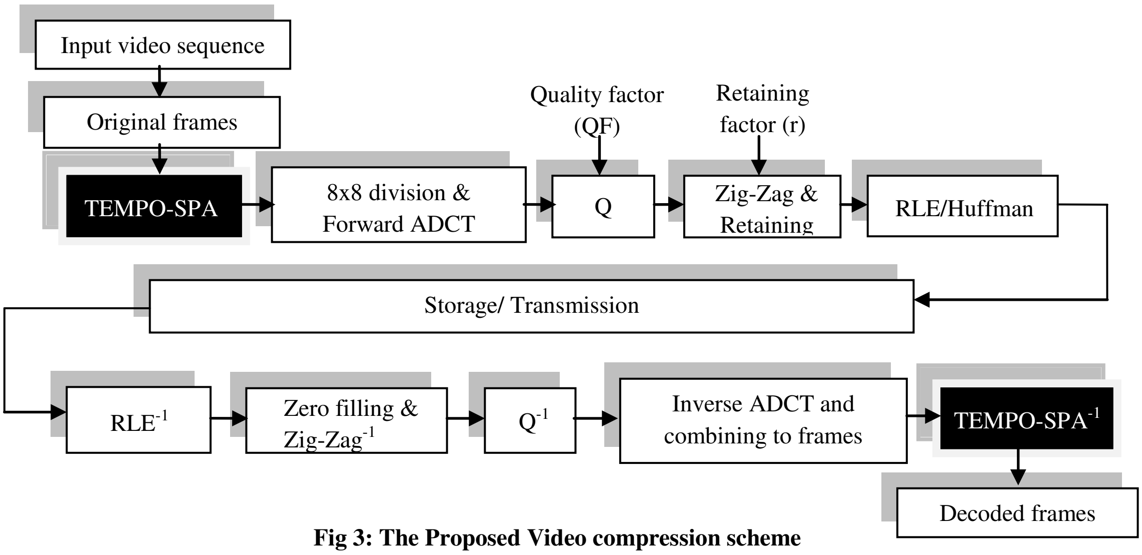


Fig 3: The Proposed Video compression scheme

3.3 Algorithm

1. Input video clip.
2. Group 8 frames of the video clip as a GOP.
3. Each GOP containing 8 frames is now TEMPO-SPA arranged to get one 2D frame from each GOP.
4. The resulting 2D frame is divided into 8x8 pixel blocks for further processing.
5. Each 8x8 pixel block is subjected to transformation using 2D-ADCT.
6. Each 8x8 transform block is quantized using standard JPEG quantization table.
7. Apply Zig-Zag scanning to convert an 8x8 matrix into a sequence of 64 coefficients.
8. Retain only the first ‘r’ DC coefficients from each of these sequences.
9. Apply RLE to these retained sequences and use any entropy coding to convert it into efficient binary form.

To decompress, the above steps are exactly repeated in the reverse order.

4. EXPERIMENTAL RESULTS

In the experiment carried out for the proposed algorithm, a standard QCIF video file of ‘mother and daughter’ is used as the input file [13]. The resolution of the video is 176x144 for Y component and 88x72 for U, V components. For experimental purpose two GOPs were taken i.e, total 16 frames were taken from each of the luminance and chrominance components.

For the first 8 frames, the 2D frame as a result of TEMPO-SPA arrangement is as shown in figure (4). From the figure we can see that this arranged frame is highly correlated and if transformed and encoded, it results with a very good compression ratio. .

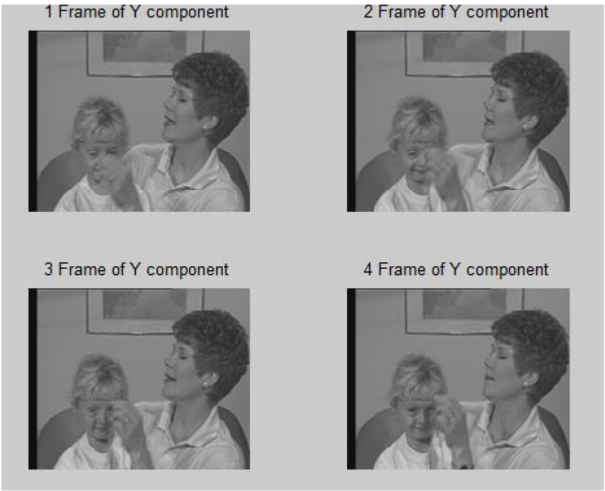


Fig 4 (a): First 4 frames of mother & daughter video

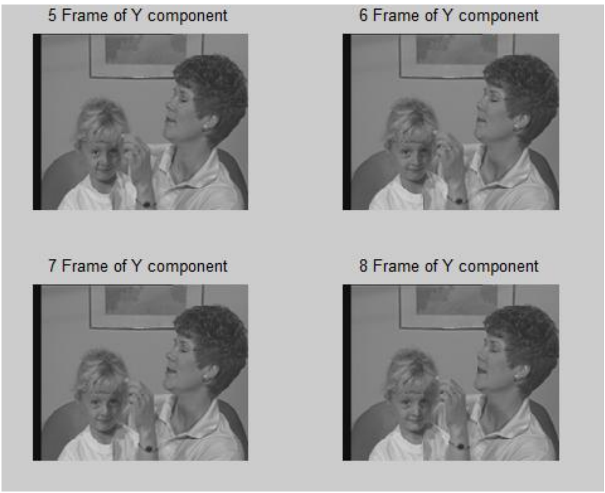


Fig 4 (b): Next 4 frames of mother & daughter video

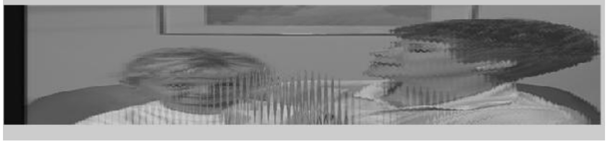


Fig 4 (c): TEMPO-SPA arrangement of above 8 frames

Table 1 shows the experimental results and observations for different quality factor QF and retaining factor ‘r’.

Table 1: Results & Observations

OS (KB)	QF	Retaining Factor (r)	CS (KB)	CR (%)	PSNR (dB)
608.3	30	5	18.3	96.9	31.82
	50	11	38	93.7	35.5
	50	30	45.8	92.4	39
	70	45	66.5	89	45.5

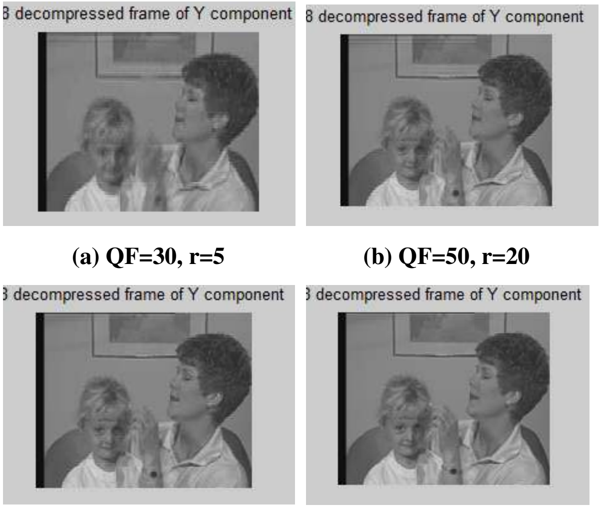
OS-original size of the video, CS-compressed size of the video, CR- compression ratio, QF- quality factor used. We can observe that for lesser retaining factor, compression is high but the PSNR or the quality of the decompressed video is very low. For QF=50 & r=30, both compression ratio as well as PSNR is optimized. In any application, if quality is at most important, then use QF=70 & r=45; which produces excellent quality and better compression ratio of about 89%.

Table 2 gives the comparison of previous algorithms and the proposed methodology.

Table 2: Comparison table

Algorithms	CR (%)	PSNR (dB)
Previous	66	45
Scalable ACC-DCT	71	42
Proposed	89	45.5

The reconstructed video frames for several different parameters are given in figure (5) [a, b, c, d].


8 decompressed frame of Y component
8 decompressed frame of Y component
3 decompressed frame of Y component
3 decompressed frame of Y component

(a) QF=30, r=5 **(b) QF=50, r=20**

(c) QF=50, r=30 **(d) QF=70, r=45**

Fig 5 (a-d): Reconstructed frames

5. CONCLUSION

In this paper, the efficient and low complex video compression scheme which uses the temporal and spatial redundancies in a video signal at its best to induce highest possible correlation in a pixel block before encoding was successfully experimented. By observing the results, it can be believed that the proposed video compression method performs well for highly correlated video sequences with less to moderate motions present and produces good rate-distortion criteria compared to other previous existing standards.

Additionally, a further development can be made to this method by using the other higher level transforms such as Wavelets, Neural networks etc, to improvise the compression ratio as well as the PSNR.

6. REFERENCES

[1] G. Suresh, P. Epsiba, Dr. M. Rajaram, Dr. S. N. Sivanandam. IJCSNS June 2010. Scalable ACC-DCT based Video compression using up/down sampling.

[2] K. Saraswathy, D. Vaithiyanathan, R. Seshasayanan. IEEE 2013. A DCT Approximation with low complexity for Image compression.

[3] M. B. T. Q. N. A. Molino, F. Vacca. Sept 2004. Low complexity video codec for mobile video conferencing.

[4] S. B. Gokturk and A. M. Aaron. (EE392J) 2002. Applying 3d methods to video for compression in Digital Video Processing.

[5] T. Fryza. Diploma Thesis 2002. Compression of Video Signal by 3D-DCT Transform.

[6] Renato J. Cintra , Fábio M. Bayer. IEEE 2011. A DCT Approximation for Image Compression.

[7] Andreas Koschan and Mongi Abidi. 2008. Digital Color Image Processing.

[8] Anil K Jain. Prentice-hall Inc 1989. Fundamentals of Digital Image Processing.

[9] G. M.P. Servais. Proc. COMSIG 1997. Video compression using the three dimensional discrete cosine transform.

[10] R. J. Cintra and V. S. Dimitrov. IEEE 2010. The arithmetic cosine transform: Exact and approximate algorithms.

[11] A. Segall and S. Lei. ISO/IEC MPEG & ITU-T VCEG 2005. Adaptive Up-sampling for Spatially Scalable Coding.

[12] G. M.P. Servais. Proc. COMSIG 1997. Video compression using the three dimensional discrete cosine transform.

[13] Mother and Daughter QCIF video file with 176 x 144. trace.eas.asu.edu/yuv/mother-daughter/mother-daughter_qcif.7z.