

3D SCENE MODELING FOR DISTRIBUTED VIDEO CODING

Matthieu Maitre*

Beckman Institute
University of Illinois at Urbana-Champaign
maitre@uiuc.edu

Christine Guillemot, Luce Morin[†]

IRISA
Rennes, France
{christine.guillemot, luce.morin}@irisa.fr

ABSTRACT

The compression efficiency of Distributed Video-Coding (DVC) suffers from the necessity of transmitting a large number of key-frames which are intra-coded. This paper describes a new 3D model-based DVC approach which reduces the key-frame frequency. The decoder first recovers a 3D model from the key-frames. It then predicts the intermediate frames by projecting it onto 2D image planes and applying image-based rendering techniques. This paper also introduces a new quasi-DVC method relying on a limited point tracking at the encoder. It greatly improves the prediction PSNR, while only slightly increasing the encoder complexity. It also allows the encoder to adaptively select the key-frames based on the video motion-content.

1. INTRODUCTION

Distributed Video Coding (DVC) is a recent approach which is being investigated as a possible alternative to classical predictive coders for applications requiring low-complexity encoders as well as error resilience. By moving the motion-compensation stage to the decoder, it keeps the encoder complexity to a minimum and benefits from compression based on both spatial and temporal correlations. Moreover, since the frames are encoded independently, it avoids temporal propagation of errors.

Previous studies [1, 2] have shown the potential of DVC but also noted the compression gap remaining between DVC and predictive coding. It is in part due to the poor performance of block-based motion-compensation when applied to distant key-frames. Key-frames need to be sent sparsely because of their high bitrate cost. This makes the motion fields between them often too complex to be approximated by spatially blockwise-constant and temporally piecewise-constant motion fields. This also requires motion vectors to be searched inside large regions, which increases the likelihood of large errors.

*The first author performed the work while at the IRISA.

[†]This work has been partly funded by the European commission in the context of the IST Network of Excellence SIMILAR

This calls for the introduction of new motion models. Unlike predictive coding, DVC does not require motion-model parameters to be sent through the communication channel, thus allowing complex models without compression penalty. In this paper, we specialize DVC to videos of static scenes obtained from a unique moving camera, a type of video of particular importance to remote exploration by drones or remote virtual reality. We take advantage of techniques developed in the context of Structure-from-Motion (SfM) [3] and propose motion models based on 3D information. First, the decoder estimates the camera parameters and the 3D scene-model from the key-frames. Then, it linearly interpolates the camera parameters at intermediate times and projects the 3D model onto the associated image planes, giving motion fields between the intermediate frames and the key-frames. Finally, it predicts the intermediate frames using Image-Based Rendering (IBR) techniques [4].

However, our experimental results shall show that even if this approach greatly improves the quality of estimated motion-fields, its impact on the PSNR of the predicted frames is limited. The prediction is hindered by the interpolation of camera-parameters at intermediate times. Therefore, we propose to go beyond DVC with an approach called quasi-DVC (qDVC). The encoder shares some limited information between frames, under the form of point tracks. This allows the decoder to estimate the camera parameters at intermediate times, instead of interpolating them. Moreover, this only slightly increases the encoder complexity. Finally, this allows the encoder to adapt the key-frame frequency to the video motion-content, a feature not possible in the DVC framework.

This article presents both DVC and quasi-DVC approaches. Section 2 describes the encoders, Section 3 details the decoders and Section 4 presents our experimental results.

2. ENCODER

2.1. Distributed video encoding

The 3D-DVC encoder is identical to a 2D-DVC encoder, as shown in Figure 1. We consider the pixel-domain codec described in [5] which improves upon the approach proposed in [1]. The DVC encoder begins by splitting the input video-

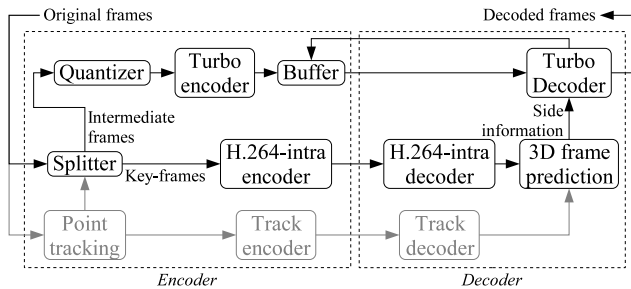


Fig. 1. Video codecs: 3D-DVC and 3D-qDVC. They differ by the point-track stream, only present in 3D-qDVC (in gray).

1 stream into key-frames and Wyner-Ziv (WZ) frames, using
 2 a constant key-frame frequency. It encodes the key-frames
 3 using a standard intra-encoder (H.264-intra in our case). It
 4 quantizes the WZ-frames into bit-planes, turbo-encodes them
 5 and transmits punctured parity-bits.

6 2.2. Quasi-distributed video encoding

7 The limitations of this purely DVC approach led us to consider
 8 an alternative quasi-DVC solution, in which a limited
 9 point-tracking is added to the encoder, as shown in Figure 1.
 10 Point-tracks offer two major benefits. First, they allow the
 11 decoder to estimate the camera parameters at intermediate
 12 times, instead of interpolating them. This greatly reduces the
 13 reprojection errors. Second, they enable the encoder to dynamically
 14 adapt the key-frame frequency: a new key-frame is
 15 only sent when the length of the longest track or the number of
 16 lost tracks exceed some thresholds. However, point-tracking
 17 also introduces overheads on the encoder complexity and the
 18 bitrate, which must be kept to a minimum.

19 We propose to attain these goals using a modified Kanade-
 20 Lucas-Tomasi (KLT) tracker [6]. First, feature points are detected
 21 on key-frames using the Harris-Stephen corner detector [7]. These points
 22 can be very sparse since the decoder only needs them to recover 11 camera
 23 parameters. Points are then tracked between consecutive frames by looking for
 24 similar intensity neighborhoods, which are assumed to follow a
 25 translational motion model. Points are robustly tracked with
 26 sub-pixel accuracy by minimizing the Sum of Squared Differences (SSD)
 27 between neighborhoods using the Levenberg-Marquardt approach [8]. The
 28 bitrate overhead is reduced by using DPCM-encoding.

31 The minimization relies on image derivatives. It is therefore
 32 limited to small motions. The range of tolerated motions is
 33 increased through a conditional multiscale approach. A basic
 34 coarse-to-fine scheme would introduce a systematic complexity
 35 overhead and miss corners too weak to be detected at coarser
 36 resolutions. Instead, the tracker first performs the SSD minimization
 37 at the finest resolution and only falls back to coarser resolutions
 38 at tracks for which it failed.

foreach pair of consecutive key-frames do

- Detect and match feature points
- Robustly estimate the fundamental matrix \mathbf{F}
- Obtain the projection matrices \mathbf{P}_0 and \mathbf{P}_1
- Triangulate the projective depths λ
- Propagate correspondences along edges
- Interpolate or estimate the projection matrices \mathbf{P}_t
- Interpolate the WZ-frames from the key-frames

Algorithm 1: Scene modeling and frame prediction

3. DECODER

40 3.1. Overview

41 As shown in Figure 1 and Algorithm 1, the decoder starts by
 42 decoding the key-frames. Then, it predicts the WZ-frames
 43 using motion interpolation. Finally, it corrects this prediction
 44 using the parity bits from the encoder and turbo-decoding.
 45 The 3D-DVC decoder differs from previous 2D-DVC decoders
 46 by its frame prediction stage. The 3D-qDVC decoder also
 47 takes advantage of point tracks to estimate the camera parameters
 48 associated with the WZ-frames, instead of interpolating them.

50 3.2. Scene modeling

51 Frame prediction takes a pair of consecutive key-frames as
 52 an input. Without loss of generality, we assume that the first
 53 key-frame was taken at time $t = 0$ and the second key-frame
 54 at time $t = 1$. Feature-points on each key-frame are denoted
 55 by respectively x_0 and x_1 . A correspondence between key-
 56 frames is denoted by (x_0, x_1) . Points are assumed to be in
 57 homogeneous coordinates. A correspondence is said to be
 58 valid when it stems from the projection of a unique 3D point X
 59 onto the image planes. That is $\exists X$ s.t. $x_0 \sim \mathbf{P}_0 X$, $x_1 \sim \mathbf{P}_1 X$
 60 where \mathbf{P}_0 and \mathbf{P}_1 are the projection matrices associated with
 61 each key-frame and ' \sim ' denotes an equality up-to-scale. This
 62 condition is equivalent to $x_1^t \mathbf{F} x_0 = 0$ where \mathbf{F} is the so-called
 63 fundamental matrix. More details can be found in [3].

64 Like at the encoder, feature-points are detected at the
 65 decoder using the Harris-Stephen corner detector. However, the
 66 number of feature-points allowed this time is much greater to
 67 obtain as many 3D points as possible. Also, the detector has
 68 now to cope with quantization noise.

69 Feature points are then matched across key-frames to obtain
 70 correspondences. A cascade of tests removes erroneous
 71 correspondences. Tests are ordered by increasing complexity
 72 so that the most complex ones handle the least correspondences.
 73 At first, correspondences between all points are considered.
 74 A first test removes correspondences whose motions are too
 75 large. A second test compares the intensity histograms of
 76 feature-point neighborhoods and removes those with poor
 77 chi-square statistics [8]. A third test proceeds similarly using
 78 the SSD as a criterion, performing a local optimization

1 of feature-point locations to obtain meaningful SSD values.
 2 A fourth test enforces that each point belongs to at most one
 3 correspondence. Finally, a fifth test removes correspondences
 4 which are not compatible with the epipolar geometry found
 5 by robustly estimating the fundamental matrix \mathbf{F} .

6 The camera parameters are estimated using self-calibration.
 7 **The interpolation of camera parameters requires the 3D space**
 8 **to be euclidean.** However, this is only possible when the cam-
 9 era motion between key-frames is generic enough, a condition
 10 rarely met in practice. Instead, we settle for quasi-euclidean
 11 self-calibration. The projection matrices can be written as
 12 $\mathbf{P}_0 = [\mathbf{I} \ 0], \mathbf{P}_1 = [\mathbf{R} \ \mathbf{t}]$ where \mathbf{I} is the identity matrix, \mathbf{R}
 13 a matrix and \mathbf{t} a vector. These quantities are related to the
 14 fundamental matrix by $\mathbf{t} \in \ker(\mathbf{F})$ and $\mathbf{R} = [\mathbf{t}]_{\times} \mathbf{F} - \mathbf{t} \mathbf{a}^t$,
 15 where \mathbf{a} is a vector and $[\cdot]_{\times}$ denotes the cross-product opera-
 16 tor. Assuming small camera rotations and slowly varying in-
 17 trinsic parameters between key-frames, the vector \mathbf{a} is found
 18 by minimizing $\|[\mathbf{t}]_{\times} \mathbf{F} - \mathbf{t} \mathbf{a}^t - \mathbf{I}\|$.

19 A cloud of 3D points is recovered by computing a pair of
 20 projective depths $\{\lambda_0, \lambda_1\}$ from each correspondence. These
 21 scalars are solutions of the equation $\lambda_1 \mathbf{x}_1 = \lambda_0 \mathbf{R} \mathbf{x}_0 + \mathbf{t}$ and
 22 are related to the underlying 3D point by $\mathbf{X} = [\lambda_0 \mathbf{x}_0^t \ 1]^t$.

23 Projective depths are only known at corners. Therefore,
 24 interpolation is required to obtain dense motion fields. Such
 25 an approximation is particularly harmful to the prediction
 26 PSNR in edge regions. Fortunately, **the intersection of edges**
 27 **and epipolar lines** gives points which, like corners, can be
 28 matched to obtain more correspondences.

29 Edge-points are detected in the first key-frame using the
 30 Canny edge-detector [3]. Correspondences are propagated by
 31 matching edge-points close to previously matched points. For
 32 a given edge-point, matching consists in a SSD-based full
 33 search along a portion of the associated epipolar line, fol-
 34 lowed by sub-pixel refinement around the best candidate via
 35 golden search [8]. The full-search domain is a small window
 36 centered around the location that the matching point would
 37 have if it followed the same motion as the one of its nearest
 38 correspondence.

39 3.3. WZ-Frame interpolation

40 Camera parameters need to be known at intermediate times
 41 to be able to project the 3D model onto **intermediate** image
 42 planes and obtain motion fields. The 3D-DVC decoder lin-
 43 earlyly interpolates them from the ones associated with the key-
 44 frames. On the other hand, the 3D-qDVC decoder estimates
 45 them from the point tracks. Using the locations of these tracks
 46 at $t = 0$ and $t = 1$, it computes the associated 3D points \mathbf{X} , as
 47 described in the previous section. Then, it obtains the projec-
 48 tion matrices \mathbf{P}_t at each intermediate time t by solving the set
 49 of equations $\mathbf{x}_t \sim \mathbf{P}_t \mathbf{X}$. The estimation approach, unlike the
 50 interpolation one, does not require the 3D space to be truly
 51 euclidean and does not assume a constant camera-motion.

52 The projection of the 3D points onto the image planes

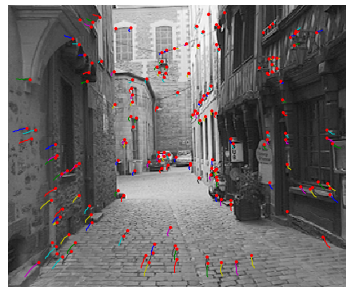


Fig. 2. Tracking (red dots: corners in first key-frame, multi-
 color curves: tracks in following WZ-frames).

53 gives motion vectors from the WZ-frames to the key-frames
 54 at corners and edges. They need to be interpolated to obtain
 55 dense motion fields. We present two interpolation schemes
 56 which differ by their assumptions on smoothness. One relies
 57 on block-matching under the epipolar geometry constraint and
 58 the other on the fitting of a mesh onto the 3D point cloud. The
 59 latter is more resistant to erroneous correspondences but tends
 60 to over-smooth depth discontinuities.

61 The epipolar block-matching scheme divides WZ-frames
 62 into blocks and searches for pairs of blocks with low SSD
 63 on the key-frames. This one-dimensional search is performed
 64 along epipolar lines in one of the key-frame, the locations in
 65 the other key-frame resulting from trifocal transfer [3]. As for
 66 the propagation along edges, full-search domains are small
 67 windows centered around locations determined by the near-
 68 est correspondences and sub-pixel accuracy is attained using
 69 golden search. Finally, each pair of blocks is linearly blended
 70 based on time to predict the WZ-frames.

71 The mesh fitting scheme estimates for each WZ-frame
 72 an elevation mesh made of regular triangles. The projec-
 73 tive depths associated with the mesh vertices are determined
 74 by energy minimization. The energy is defined as the depth
 75 distance between the mesh and the 3D points along with a
 76 Tikhonov regularization. Correspondences which lead to in-
 77 verted triangles or large depth errors are removed. This mesh
 78 is projected onto the key-frames, which are then warped using
 79 2D texture mapping and blended to predict the WZ-frame.

80 4. EXPERIMENTAL RESULTS

81 **The codecs were evaluated on several sequences.** We present
 82 **here the** results on the 50 first frames of the street sequence,
 83 a CIF sequence at 30fps. The camera is mostly moving for-
 84 ward, with some slight rotations. Figure 2 displays its first
 85 frame, along with the points-tracks between the first two key-
 86 frames. Note their sparseness. The bitrate overhead intro-
 87 duced by point-tracking is .01b/sample. The encoder-com-
 88 plexity overhead is negligible compared to predictive 3D vi-
 89 deo-encoding [9], and 35 times smaller than basic 2D block-
 90 matching with an integer search-range of equivalent size. The

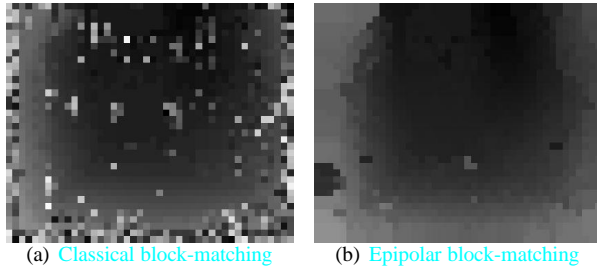


Fig. 3. Norm of the block motion-fields between the first two key-frames (same intensity scaling)..

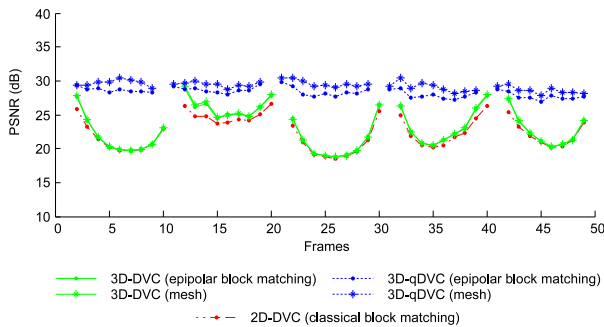


Fig. 4. PSNR of interpolated WZ-frames using lossless key-frames.

1 adaptative key-frame frequency induced by point-tracks is ap-
 2 proximately one key-frame every 10 frames. To allow com-
 3 parison, the key-frame frequencies of 2D-DVC, 3D-DVC and
 4 H.264-inter were set to the same value.

5 Figure 3 compares the motion-fields obtained with clas-
 6 sical block-matching and with the proposed epipolar block
 7 matching. The latter is qualitatively superior, the number
 8 and size of errors being much smaller. Figure 4 compares
 9 the PSNR of the WZ-frame interpolation by 2D-DVC, 3D-
 10 DVC and 3D-qDVC using lossless key-frames. The qualita-
 11 tive improvement of motion-fields has no significant effect.
 12 However, point-tracking increases the PSNR by up to 10dB.
 13 Figure 5 compares the rate-distortion performances of H.264-
 14 intra, H.264-inter (IP..PI), the 2D-DVC Discover codec [5]
 15 and 3D-qDVC. Our codec outperforms both 2D-DVC and
 16 H.264-intra, and approaches H.264-inter at lower bitrates.

5. CONCLUSION

17
 18 In this paper we proposed new DVC methods based on 3D re-
 19 construction. We showed that the epipolar geometry helps im-
 20 proving the quality of motion-fields. Moreover, adding point-
 21 tracking to the encoder significantly increases the prediction
 22 PSNR and allows adaptive key-frame frequencies, while only
 23 introducing limited overheads. Future work shall consider the
 24 non-i.i.d nature of prediction errors to improve turbo-decoding.

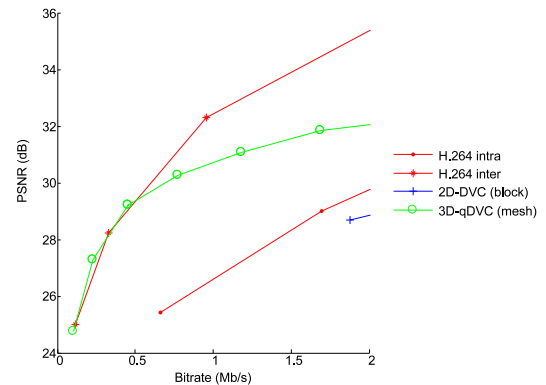


Fig. 5. Rate distortion for H.264-intra, H.264-inter, 2D-DVC and 3D-qDVC, using lossy key-frames.

6. ACKNOWLEDGMENT

26
 27 We are thankful to both the IST development team and the
 28 Discover software team for providing the 2D-DVC codec.

7. REFERENCES

- 29
 30 [1] B. Girod, A.M. Aaron, S. Rane, and D. Rebollo-
 31 Monedero, “Distributed video coding,” *Proc. of the*
 32 *IEEE*, vol. 93, no. 1, pp. 71–83, January 2005.
 33 [2] R. Purit and K. Ramchandran, “PRISM: A new robust
 34 video coding architecture based on distributed compres-
 35 sion principles,” in *Proc. Allerton Conf.*, 2002.
 36 [3] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern*
 37 *Approach*, Prentice Hall, August 2002.
 38 [4] H.Y. Shum and S.B. Kang, “A review of image-based ren-
 39 dering techniques,” in *Proc. SPIE Conf. on Visual Com.*
 40 *and Im. Proc.*, 2000.
 41 [5] J. Ascenso, C. Brites, and F. Pereira, “Improving frame
 42 interpolation with spatial motion smoothing for pixel do-
 43 main distributed video coding,” in *EURASIP Conf. on*
 44 *SIPMCS*, 2005.
 45 [6] S. Baker and I. Matthews, “Lucas-Kanade 20 years on: a
 46 unifying framework,” *IJCV*, vol. 56, no. 3, pp. 221–255,
 47 2004.
 48 [7] C. Harris and M. Stephens, “A combined corner and edge
 49 detector,” in *Proc. Alvey Vision Conf.*, 1988, pp. 147–151.
 50 [8] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vet-
 51 terling, *Numerical Recipes in C : The Art of Scientific*
 52 *Computing*, Cambridge University Press, 1993.
 53 [9] R. Balter, P. Gioia, and L. Morin, “Time evolving 3D
 54 model representation for scalable video coding,” in *Proc.*
 55 *ICIP*, 2005.