Proc. 32nd Picture Coding Symposium (PCS 2016, Nuremberg, Germany, December 2016).
The final publication will be available on IEEE Xplore.

A Proof-of-Concept Framework for PDE-Based Video Compression

Sarah Andris, Pascal Peter, and Joachim Weickert
Mathematical Image Analysis Group, Faculty of Mathematics and Computer Science
Campus E1.7, Saarland University, 66041, Saarbrücken, Germany
{andris, peter, weickert}@mia.uni-saarland.de

***Abstract*—In image compression, codecs that rely on interpolation with partial differential equations (PDEs) are becoming increasingly popular. However, there have not been many attempts to transfer this concept to video compression. Since real-time performance is challenging for PDE-based reconstruction, first efficient approaches work on a frame-by-frame basis and focus on parallel implementations without considering coding quality. So far, there is no fully PDE-based video codec that exploits temporal redundancies. As a remedy, we propose a modular framework that combines PDE-based compression with motion compensation: Intra frames are predicted with PDE-based inpainting and inter frames with dense optic flow fields. We use this framework to develop a proof-of-concept codec that combines homogeneous diffusion inpainting with the variational optic flow model of Brox et al. (2004). Even without sophisticated parallelisation, we are able to perform real-time decompression of colour videos for the first time in PDE-based video compression.**

I. Introduction

Video compression is more popular than ever due to online streaming services, resulting in vivid research in this field. The most popular methods belong to the MPEG family [1]. They rely on a combination of prediction and transform coding of difference images: So called *intra frames* are predicted without any reference to other frames, while *inter frames* are estimated using preceding or subsequent frames and a motion field between them.

We can interpret intra frame prediction as image compression. Here, PDE-based inpainting methods have increasingly attracted interest. They exploit sparsity in the spatial rather than the transform domain by storing only a fraction of all pixel values and reconstructing missing parts with inpainting. The current state-of-the-art methods in this field for grey value images by Schmaltz et al. [2] and an adapted version for colour images by Peter and Weickert [3] can beat JPEG 2000 [4].

While there have been first attempts towards video compression with PDEs [5]–[8], these codecs work on a strict frame-by-frame basis. Thus, they do not exploit temporal redundancies.

A. Our Contribution

We aim at taking the first step towards PDE-based video codecs that integrate popular ideas from established transform-based methods such as motion-based inter frame prediction. We put special effort into designing a general framework that can be used as a basis for future research in this area.

The framework is divided into several modules that each represents a specific part of video compression and can be equipped with different methods. There are three main modules: partitioning into groups of pictures (GOPs), prediction of frames either through inpainting or motion compensation, and storage of all data with a suitable encoding.

Furthermore, we propose a first proof-of-concept video codec for this framework. We want to show that our framework can be used to acquire codecs with real-time performance on the decoder side by choosing relatively simple and well-understood ingredients for the different modules. To this end, we employ homogeneous diffusion for the inpainting part and use dense backward optic flow fields (BOFFs) for motion compensated prediction generated by the method of Brox et al. [9]. These BOFFs contain the displacements of all pixels in a frame $\boldsymbol{u}_{k+1}$ at time step $k+1$ w.r.t. its preceding frame $\boldsymbol{u}_k$. Finally, we encode all generated data with adaptive arithmetic coding [10] and bzip2 [11].

B. Related Work

For an overview of the basic concepts of MPEG and in particular its latest standard H.264/AVC [12], we refer the reader to the survey by Sullivan and Wiegand [13].

PDE-based video compression has not yet received a lot of attention so far. Schmaltz and Weickert [14] have proposed a specialised codec that compresses only the static background with PDE-based techniques. Köstler et al. [5] have presented an approach for real-time playback on a *Playstation 3* with a CELL multicore processor. First steps to expand the R-EED codec by Schmaltz et al. [2] to real-time video compression have been taken by Baum [6] and have been extended by Peter et al. [7]. However, these algorithms rely only on a frame-by-frame compression and do not take redundancies in time direction into account. They focus more on showing that inpainting in each frame is possible in real-time with advanced parallelised algorithms. Furthermore, these coders do not provide a mode for colour videos.

Gao [8] has used PDE-based compression on motion fields. However, the remaining parts of his codec are still transform-based, and he has only been able to reach real-time performance in low bit rate scenarios.

There are approaches which incorporate PDE-based ideas into the H.264/AVC standard. Liu et al. [15] have used homogeneous inpainting together with edge information for intra frame prediction. In a similar direction, Doshkov et al. [16] have combined homogeneous inpainting with template matching for the same task.

Sullivan and Wiegand [13] note in their survey that the most significant improvements in video compression were due to more sophisticated methods in the motion compensated prediction. Therefore, we want to allow including optic flow methods that provide accurate, dense displacement vector fields between subsequent frames in our framework.

Moulin et al. [17] as well as Han and Podilchuk [18] have employed dense optic flow fields for motion compensated prediction of inter frames. Both approaches optimise their methods with respect to the coding efficiency of the resulting residual. More recently, Chen and Mied [19] used a block based representation of their motion fields using bilinear polynomial functions. Although they have not given a rigorous comparison of compression ratios, they showed that their predictions were more accurate than standard block matching.

C. Paper Structure

In Section II we propose our framework for PDE-based video compression. It acts as a basis for a proof-of-concept codec in Section III. The corresponding experiments are shown in Section IV, and we conclude our work in Section V.

II. Framework for PDE-Based Video Compression

In the following, we present a framework for video compression that combines the concepts of intra and inter coding with PDE-based approaches: First, we predict intra frames with PDE-based inpainting from a small amount of stored image pixels. These frames are self-contained and can thus be used for random access. In contrast, inter frames use motion compensation based on preceding or subsequent frames. We compensate the resulting errors of both prediction types with residual images.

The framework is divided into self-contained parts allowing us to change methods within these parts easily. This modularity is important for future work on PDE-based video compression, since we can examine the influence of different methods without having to reimplement other parts. To give a better overview over the submodules of our framework, we divide them into three main modules. The first module deals with the partitioning of the video into GOPs. The second module handles all submodules that engage in the prediction of frames. Finally, the third module contains the storage and encoding of the data that has been produced in the second module. Figure 1 provides an overview of the interactions between all submodules.

A. Partitioning

We assume that our input videos contain semantically coherent scenes. Using *scene detection* to partition a video into GOPs is advantageous for several reasons. We can treat each GOP individually, resulting in a reduced computational effort and lower memory consumption. Furthermore, it allows to adapt our prediction to the local structure of each individual

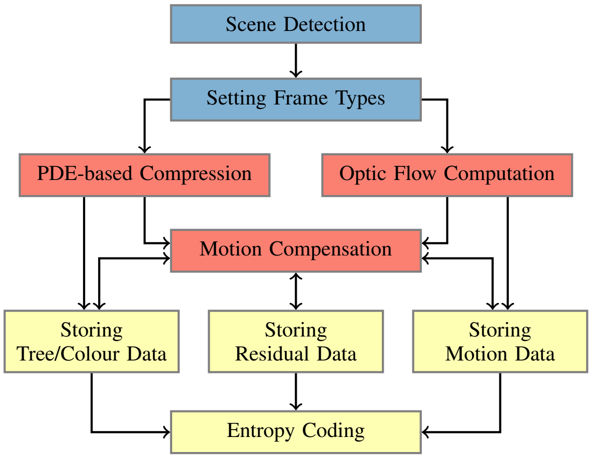


Fig. 1: Structure of the proposed framework.

scene. Detecting scenes often correlates with *setting frame types*: A scene change indicates to set an intra frame, as we cannot predict it reliably with motion compensation. Since this affects the first frame of a GOP, this is also useful for random access. All other frames within one scene are usually inter frames, possibly belonging to subtypes for different prediction methods. Long contiguous scenes can also be separated into multiple GOPs. Individual codecs define the exact criteria for type setting depending on the prediction modules used.

B. Prediction

Since prediction is the core module of our framework, codec performance depends significantly on its submodules. Our intra frame prediction employs *PDE-based compression*: We only store a small fraction of pixels and reconstruct the missing image parts with inpainting. Let $\boldsymbol{f} : \Omega \to \mathbb{R}^3$ be a colour image that maps positions $\boldsymbol{x}$ in the image domain Ω to RGB values. Colour values are only known for the *inpainting mask* $K \subset \Omega$. We reconstruct the missing values in $\Omega \setminus K$ by solving the general inpainting problem

$$\partial_t u_c = L u_c \qquad \text{on } \Omega \setminus K \times (0, \infty), \qquad (1)$$

$$u_c(\boldsymbol{x}, t) = f_c(\boldsymbol{x}) \qquad \text{on } K \times [0, \infty). \qquad (2)$$

Here, u_c with $c = \{R, G, B\}$ are the colour channels of the evolving image $\boldsymbol{u}(\boldsymbol{x}, t)$ with time parameter t, and L is a suitable differential operator. Furthermore, we impose reflecting boundary conditions on $\partial\Omega$. Known colour values stay fixed (2), while the steady state $(t \to \infty)$ of the evolution in (1) yields the restored image in the unknown regions. The choice of L and K influences the inpainting performance significantly and is an important element of codec design.

For inter frames, our module for dense *optic flow computation* provides displacement vector fields with floating point precision on a per-pixel basis. There are two types of these fields: Backward optic flow fields (BOFFs) give displacements from a frame $\boldsymbol{u}_{k+1}$ to its preceding frame $\boldsymbol{u}_k$ and are used for forward prediction from $\boldsymbol{u}_k$ to $\boldsymbol{u}_{k+1}$. Forward optic flow fields (FOFFs) are used for the opposite direction. With this information, the *motion compensation* module computes predictions.

We acquire the corresponding residuals by comparing the predictions with the original data. Note that during decoding, only the stored motion fields and residuals are available. Therefore, we have to incorporate potential lossy compression steps in the storage module into the residual computation to avoid error propagation.

C. Storage

Finally, all data required for frame reconstruction need to be stored efficiently. Individual modules for storing *tree and colour*, *optic flow*, and *residual* data employ lossy compression techniques such as quantisation. An efficient codec needs to apply adequate strategies for each kind of data. In a final step, we remove remaining redundancies with lossless *entropy encoding*. Note that omitting lossy steps in the residual storage module leads to fully lossless video compression.

III. A Proof-of-Concept Codec

In order to show that the proposed framework can be used effectively as a basis for new PDE-based video codecs, we implement a proof-of-concept codec. We want to show that such a codec is able to compress video data by a reasonable amount and can decode compressed videos in real-time, even if only basic methods are used within each module. In the following, we describe our design choices for every module.

A. Encoder

We combine scene detection and the identification of frame types by setting a new GOP as soon as we encounter an intra frame. All subsequent frames are inter frames which we predict with the directly preceding frame and a corresponding BOFF. As a criterion when to use an intra frame, we use the root mean square error (RMSE) between subsequent frames.

For PDE-based intra frame prediction, we use the simplest possible differential operator L: Choosing the Laplacian operator results in *homogeneous diffusion inpainting* [20]. We get the evolution

$$\partial_t u_c = \Delta u_c \quad \text{on } \Omega \setminus K \times (0,\infty) \tag{3}$$

for the propagation of values in unknown regions. Homogeneous diffusion does not have any additional parameters that have to be optimised and there exist highly efficient solvers. In contrast to Peter et al. [7], we do not require heavy parallelisation on GPUs and can use the sequential multigrid solver by Mainberger et al. [21].

For the selection of the inpainting mask we choose the approach of Schmaltz et al. [2], which allows mask points only at grid positions of a rectangular subdivision of a frame. We adapt the inpainting mask to the image structure by adding mask points at locations with high local error after inpainting. Since we can describe these locations based on the underlying rectangular subdivision, the mask is coded as a binary tree.

For each GOP, we compute a dense BOFF which contains displacement vectors in a frame pointing to the previous frame. We use an implementation [22] of the method of Brox et al. [9]. Since this is a variational method, the flow computation is also PDE-based. Afterwards, we subsample the flow fields coarsely. This high quality approach reduces coding costs and allows reconstruction in real-time. The size of the blocks in the acquired flow fields is a free parameter and influences the compression rate in two different ways. On one hand, smaller blocks yield more accurate predictions and thus the residuals are easier to compress. On the other hand, smaller blocks mean more data to store for each flow field. Afterwards we transform the displacements channelwise to the range $[0, 255]$ with an affine transformation that can be adapted to the minimal and maximal values in the flow field. Thus, we can store the motion vectors with two bytes per block.

We use the acquired displacement fields for inter frame prediction via motion compensation. Let us assume that we have two consecutive discrete frames $\boldsymbol{u}_k$ and $\boldsymbol{u}_{k+1}$ and their BOFF $(v_k, w_k)^\top$ that gives the displacement from frame $\boldsymbol{u}_{k+1}$ to $\boldsymbol{u}_k$. Then we predict $\boldsymbol{u}_{k+1}$ in (i, j) with

$$\boldsymbol{u}_{k+1,i,j} = \boldsymbol{u}_{k,\tilde{i},\tilde{j}} \quad \text{with} \quad \tilde{i} = i + v_{i,j},\, \tilde{j} = j + w_{i,j}\,, \tag{4}$$

assuming that the grid sizes are 1. Since in general $v_{i,j}, w_{i,j} \notin \mathbb{Z}$, we have to approximate $\boldsymbol{u}_{k,\tilde{i},\tilde{j}}$. To this end, we compute a weighted average of the four grid neighbours around the intergrid position $(\tilde{i}, \tilde{j})$.

Afterwards we immediately acquire the corresponding residual as the difference to the original frame. We shift the residual such that all values lie in $[0, 255]$ and then quantise the resulting values. Since we can expect that most values are predicted with high accuracy, we assume that the shifted values lie close to 128. Therefore, we use a finer quantisation in the vicinity of 128 for higher quality. For every frame, we compute the reconstruction that the decoder will produce with the available data and use this frame for further motion compensated predictions. This way, errors due to lossy storage of data are not amplified within one GOP.

As soon as we reach the end of a GOP, we store colour values and tree data for the intra frame, displacement vectors for the inter frames, and residuals for both types. Adaptive arithmetic coding provides a good trade-off between coding efficiency and runtime for the colour values and motion vectors. Since the residual is not sparse and we expect very long runs of the same value we employ bzip2 which combines run length encoding with fast performance. Finally, we concatenate all files to the final output file. After we have encoded the last GOP, we generate the global header which contains all information necessary for decoding.

B. Decoder

The main method of the decoder first reads the global header information and then starts three threads containing entropy decoding, the reconstruction of frames, and playback. These threads share data structures for decoded data and fully reconstructed frames and communicate via global counters. The reconstruction method checks if decoded data is available and if the cache for playback is not full. If this is the case, we gather all required data, build the inpainting mask, reverse the quantisation applied by the encoder for colour value and residual data and transform motion and residual data back to their original representation. Then we either perform inpainting or motion compensation depending on the type of the current frame (intra or inter). Afterwards we subtract the residual to correct the error.

IV. Experiments

We run our experiments on an *Intel Xeon CPU W3565@3.20GHz*. As a test set we use a sequence of 1000 frames of the well-known *Sintel* video by Rosendaal [23] in a resolution of 854×364. This short film incorporates all aspects on which we want to test our codec: It is a colour movie, contains a lot of textured areas, as well as complicated motion.

We want to compare our results with the fully PDE-based video codec by Peter et al. [7]. It employs R-EED image compression [2] on each frame with some additional overhead such as GOP information. However, this codec does not have a colour mode. Therefore, we exchange the R-EED frame compression method with the current state of the art in PDE-based colour image compression: R-EED-LP by Peter and Weickert [3].

We choose the compression ratio of the inpainting-based intra frame prediction to be $10 : 1$ and find an optimal quantisation with respect to the MSE. We only use 1% of all motion vectors in the BOFFs and set the residual quantisation parameter to $q = 16$. The resulting compression rate is 10:1, which we also use the R-EED-LP video codec.

Figure 2 shows two original frames of *Sintel* [23]. We predict frame 4642 as an intra frame and 4643 as an inter frame. Correspondingly, we show the inpainting mask, BOFF, and residuals. Since a value of 128 means an error of 0, the residual image is mostly grey. The contrast enhanced zooms shows a typical region of errors: Inpainting struggles with texture, while motion compensation has problems at motion boundaries where we can expect problems with occlusions or disocclusions.

Figure 3 compares the results of our method with the R-EED-LP video codec. We are able to consistently reach higher quality in reconstructed frames at the same compression rate. For the currently used set with 1000 frames, the overall MSE of the R-EED-LP video codec is 28.3, while we achieve an MSE of 12.0. This is particularly remarkable, since we did not use advanced concepts such as anisotropic nonlinear diffusion or YCbCr colour space representations. This indicates that our proof-of-concept approach still has a high potential for further improvements.

V. Conclusion and Outlook

We introduced a modular framework for PDE-based video compression that supports a wide variety of different methods for inpainting, motion compensation, and encoding. As a proof-of-concept, we implemented the first fully PDE-based video codec that exploits temporal redundancy. The core idea in our approach is to employ inpainting and motion compensation as prediction mechanisms. Since we use mostly inter frames that can bedecoded very fast with motion compensation, we can reach real-time performance on colour videos without advanced parallelised algorithms for inpainting. Furthermore, using inpainting only for prediction enables us to compensate resulting errors. Compared to a pure frame-by-frame codec with PDEs, our incorporation of temporal redundancy reduced the MSE by a factor 2.3.

The modularity of our framework allows us to improve our codec by simply substituting the methods in the different parts. Our ongoing work includes the introduction of new inter frame types, more advanced techniques in the inpainting and optic flow computation, and exploring different possibilities to store and encode motion and residual data.

References

[1] ISO/IEC, "Information technology – coding of audio-visual objects – part 10: Advanced video coding," 2003, standard, ISO/IEC 14496-10.

[2] C. Schmaltz, P. Peter, M. Mainberger, F. Ebel, J. Weickert, and A. Bruhn, "Understanding, optimising, and extending data compression with anisotropic diffusion," *International Journal of Computer Vision*, vol. 108, no. 3, pp. 222–240, Jul. 2014.

[3] P. Peter and J. Weickert, "Colour image compression with anisotropic diffusion," in *Proc. 21st IEEE International Conference on Image Processing*, Paris, France, Oct. 2014, pp. 4822–4862.

[4] D. S. Taubman and M. W. Marcellin, Eds., *JPEG 2000: Image Compression Fundamentals, Standards and Practice*. Boston: Kluwer, 2002.

[5] H. Köstler, M. Stürmer, C. Freundl, and U. Rüde, "PDE based video compression in real time," Lehrstuhl für Informatik 10, University Erlangen–Nürnberg, Germany, Tech. Rep. 07-11, 2007.

[6] K. Baum, "GPGPU supported diffusion-based naive video compression," Master Thesis, Department of Computer Science, Saarland University, Saarbrücken, Germany, 2013.

[7] P. Peter, C. Schmaltz, N. Mach, M. Mainberger, and J. Weickert, "Beyond pure quality: Progressive modes, region of interest coding, and real time video decoding for PDE-based image compression." *Journal of Visual Communication and Image Representation*, vol. 31, no. 4, pp. 253–265, Aug. 2015.

[8] Q. Gao, "Low bit rate video compression using inpainting PDEs and optic flow," Master Thesis, Department of Computer Science, Saarland University, Saarbrücken, Germany, 2008.

[9] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in *Computer Vision-ECCV 2004*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Berlin: Springer, 2004, vol. 8689, pp. 25–36.

[10] J. J. Rissanen, "Generalized Kraft inequality and arithmetic coding," *IBM Journal of Research and Development*, vol. 20, no. 3, pp. 198–203, May 1976.

[11] J. Seward, "bzip2 and libbzip2, version 1.0.5, a program and library for data compression," 2008. [Online]. Available: http://www.bzip.org/.

[12] ITU-T and ISO/IEC JTC 1, "Advanced video coding for generic audio-visual services, ITU-T recommendation H.264 and ISO/IEC 14496-10 (MPEG-4 AVC)," Apr. 2007, standard version 7.

[13] G. J. Sullivan and T. Wiegand, "Video compression – From concepts to the H. 264/AVC standard," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 18–31, Jan. 2005.

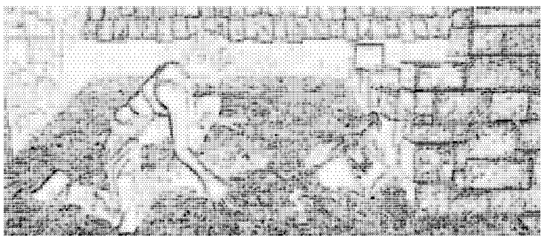
[14] C. Schmaltz and J. Weickert, "Video compression with 3-D pose tracking, PDE-based image coding, and electrostatic halftoning," in *Pattern Recognition*, ser. Lecture Notes in Computer Science, A. Pinz, T. Pock, H. Bischof, and F. Leberl, Eds. Berlin: Springer, 2012, vol. 7476, pp. 438–447.

[15] D. Liu, X. Sun, F. Wu, and Y.-Q. Zhang, "Edge-oriented uniform intra prediction," *IEEE Transactions on Image Processing*, vol. 17, no. 10, pp. 1827–1836, Oct. 2008.

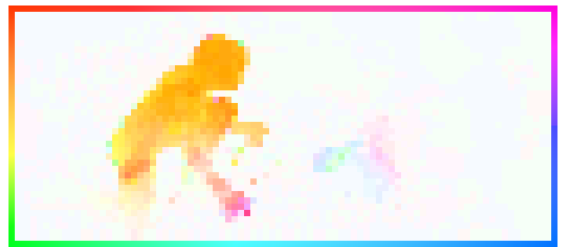
[16] D. Doshkov, P. Ndjiki-Nya, H. Lakshman, M. Koppel, and T. Wiegand, "Towards efficient intra prediction based on image inpainting methods," in *Proc. Picture Coding Symposium*, Nagoya, Japan, Dec. 2010, pp. 470–473.

(a) Original (intra) frame 4642.

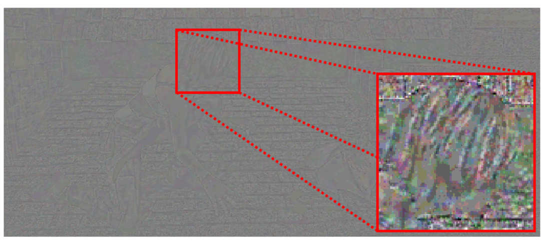
(b) Original (inter) frame 4643.



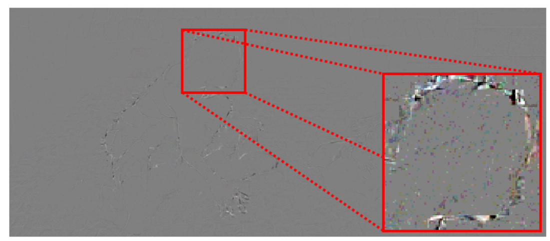
(c) Inpainting mask.



(d) BOFF.

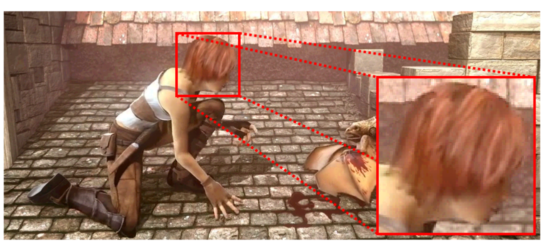


(e) Residual for (intra) frame 4642.

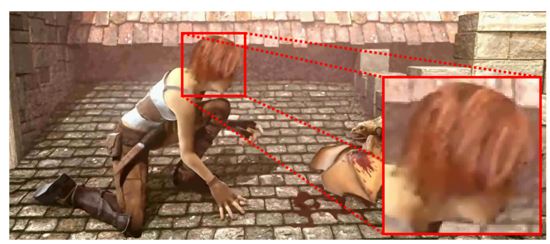


(f) Residual for (inter) frame 4643.

Fig. 2: Intermediate results of our video codec at a compression rate of 10:1 with contrast enhanced zooms. The colour coding of the BOFF is adapted from [24].



(a) Our method. MSE = 12.0



(b) R-EED-LP video codec. MSE = 28.2

Fig. 3: Reconstruction of frame 4643 with our video codec and the R-EED-LP codec at compression rate 10:1.

[17] P. Moulin, R. Krishnamurthy, and J. W. Woods, "Multiscale modeling and estimation of motion fields for video coding," *IEEE Transactions on Image Processing*, vol. 6, no. 12, pp. 1606–1620, Dec. 1997.

[18] S.-C. Han and C. Podilchuk, "Video compression with dense motion fields," *IEEE Transactions on Image Processing*, vol. 10, no. 11, pp. 1605–1612, Nov. 2001.

[19] W. Chen and R. Mied, "Optical flow estimation for motion-compensated compression," *Image and Vision Computing*, vol. 31, no. 3, pp. 275–289, Jan. 2013.

[20] S. Carlsson, "Sketch based coding of grey level images," *Signal Processing*, vol. 15, no. 1, pp. 57–83, Jul. 1988.

[21] M. Mainberger, A. Bruhn, J. Weickert, and S. Forchhammer, "Edge-based compression of cartoon-like images with homogeneous diffusion," *Pattern Recognition*, vol. 44, no. 9, pp. 1859–1873, Sep. 2011.

[22] J. Sánchez, N. Monzón, and A. Salgado, "Robust optical flow estimation," *Image Processing Online*, vol. 3, pp. 252–270, Oct. 2013.

[23] T. Roosendaal, "Sintel," in *ACM SIGGRAPH 2011 Computer Animation Festival*, New York, NY, USA, 2011, p. 71.

[24] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," *International Journal of Computer Vision*, vol. 92, no. 1, pp. 1–31, Oct. 2011.