Proc. 26th IEEE International Conference on Image Processing
(ICIP 2019, Taipei, Taiwan, September 2019), 3557-3561, 2019.
The final publication is available on IEEE Xplore.

FAST INPAINTING-BASED COMPRESSION: COMBINING SHEPARD INTERPOLATION WITH JOINT INPAINTING AND PREDICTION

Pascal Peter

Mathematical Image Analysis Group, Faculty of Mathematics and Computer Science,
Campus E1.7, Saarland University, 66041 Saarbrücken, Germany.
peter@mia.uni-saarland.de

ABSTRACT

Inpainting-based compression has been suggested as a qualitative competitor to the JPEG family of transform-based codecs, specifically for high compression ratios. However, it also requires sophisticated interpolation, data optimisation and encoding tasks that are both slow and hard to implement. We propose a fast and simple alternative that combines Shepard interpolation with a novel joint inpainting and prediction approach. It represents the image by a fraction of its pixel values on a sparse regular subgrid that are selected by an efficient optimisation strategy. Experiments show that our codec is up to five orders of magnitude faster than traditional inpainting-based approaches. Qualitatively, it can surpass transform-based codecs, in particular for high compression ratios and cartoon-like images.

***Index Terms*—** Image compression, inpainting, prediction

1. INTRODUCTION

Lossy inpainting-based image compression [1, 2] relies on the principle of spatial sparseness: Codecs select and store a small fraction of image points. During decompression, a suitable interpolation operator restores the missing data based on this known pixel mask, thus performing inpainting. Qualitatively, these codecs can compete with JPEG [3] and JPEG 2000 [4], in particular for piecewise smooth images [5, 6] and high compression ratios on natural images [2].

So far, quality was the focus of research on inpainting-based compression, while runtime was mostly neglected. Real-time decompression has been considered for video decoding [7, 8, 9], but fast encoding has not been a dedicated research goal. Even real-time decoding already constitutes a significant challenge since current inpainting-based codecs consist of sophisticated and time-intensive building blocks. They supplement advanced inpainting methods [2] with cost-intensive strategies for the optimisation of known data [2, 8]. Therefore, fast decoding without loss of quality has been only achieved with state-of-the-art numerics, as well as GPU and CPU parallelisation. Additionally, many inpainting-based codecs use PAQ [10] for encoding, an efficient but slow context-mixing approach that involves predictions by a large number of neural networks. Overall, current codecs suffer from two drawbacks of inherent complexity: Slow encoding speed and non-trivial implementation.

Our contribution. We propose a proof-of-concept codec for fast inpainting-based image compression with simple ingredients. Our framework aims at reducing complexity while preserving the strengths of inpainting-based compression: Good quality for high compression ratios and cartoon-like images. To this end, we replace advanced inpainting and optimisation techniques by simple Shepard interpolation [11, 12] on a regular grid of known pixels. This removes storage costs for positional data and can be combined with a novel fast optimisation strategy for the corresponding grey values. For encoding, we combine finite state entropy (FSE) [13, 14], a fast alternative to arithmetic coding, with a new concept: *joint inpainting and prediction* (JIP) estimates values from partial pixel masks during inpainting at negligible additional cost. We compare our new strategy to compression with homogeneous diffusion inpainting and to the transform-based codecs JPEG and JPEG 2000.

Related work. For image reconstruction, we rely on Shepard interpolation [11] that restores missing pixels as a normalised average of known data weighted by an inverse distance function. More recent variants of this method include normalised convolution [15] of Knutsson and Westin,

Part of our research has been funded by the ERC Advanced Grant INCOVID. This is gratefully acknowledged.

as well as the efficient filtering by adaptive normalisation (EFAN) of Achanta et al. [12].

Codecs with homogeneous diffusion [5, 6, 16, 17, 18, 19] resemble our approach since they use a simple inpainting operator. However, they rely either on more sophisticated data selection or encoding strategies. Real-time performance has been only considered for video coding [7, 8, 9] and relies on complex numerics, not on simplicity. The methods of Galić et al. [1] and Schmaltz et al. [2] use more advanced inpainting, but also inspire our new strategy for grey value optimisation.

Organisation of the paper. Section 2 describes relevant inpainting techniques. We introduce our new codec in Section 3 and evaluate it in Section 4. The paper concludes with a summary and discussion of future work in Section 5.

2. FAST AND SIMPLE INPAINTING

Our codec relies on Shepard interpolation [11, 12, 15]. It is not only fast, but also allows us to design novel prediction and optimisation components for our compression pipeline in Section 3. For the sake of comparison we also consider homogeneous diffusion inpainting as the most widely-used operator in inpainting-based compression [5, 6, 16, 17, 18, 19].

2.1. Shepard interpolation

Consider a discrete image $\boldsymbol{f} \in \mathbb{R}^{m \times n}$ which is only known on the inpainting mask, a subset K of the image domain $\Omega = \{1, ..., m\} \times \{1, ..., n\}$. For $\boldsymbol{x}_i \in \Omega$, we can compute a reconstruction $u_i := u(\boldsymbol{x}_i)$ by averaging known data weighted by a function w:

$$u_i = \frac{\sum_{\boldsymbol{x}_j \in K} w(\boldsymbol{x}_j - \boldsymbol{x}_i) f_j}{\sum_{\boldsymbol{x}_j \in K} w(\boldsymbol{x}_j - \boldsymbol{x}_i)}. \tag{1}$$

The EFAN variant [12] of this general Shepard interpolation approach defines w by a truncated Gaussian $G(\boldsymbol{x}) := \exp((-x_1^2 - x_2^2)/(2\sigma^2))$ with standard deviation σ and size $(\lceil 4\sigma \rceil + 1) \times (\lceil 4\sigma \rceil + 1)$. In Eq. (1), the Gaussian is centred at the point $\boldsymbol{x}_i$. The computation of a full image reconstruction operations and is linear in the number of pixels $m \cdot n$.

The only parameter of the algorithm, the standard deviation σ of the Gaussian, can be automatically adapted to the mask density by $\sigma = \sqrt{(mn)/(\pi |K|)}$. We have experimentally confirmed this as a good choice for our purposes.

2.2. Inpainting with homogeneous diffusion

In our experiments, we also consider inpainting based on homogeneous diffusion [20]. We write the original image as a vector $\boldsymbol{f} \in \mathbb{R}^{mn}$ by traversing it row by row. Furthermore, we express the set of known data K by a diagonal matrix $\boldsymbol{C} \in \mathbb{R}^{mn \times mn}$: Its entries $c_{i,i}$ are 1 for known pixels f_i and 0 otherwise. Then a diffusion-based inpainting result is the solution $\boldsymbol{u}$ of the inpainting equation

$$\boldsymbol{C}(\boldsymbol{u} - \boldsymbol{f}) - (\boldsymbol{I} - \boldsymbol{C})\boldsymbol{L}\boldsymbol{u} = \boldsymbol{0}, \tag{2}$$

where $\boldsymbol{I} \in \mathbb{R}^{mn \times mn}$ is the unit matrix, and $\boldsymbol{L}\boldsymbol{u}$ is a discretisation of the spatial Laplacian $\Delta u = \partial_{xx} u + \partial_{yy} u$ with reflecting boundary conditions. The first term of Eq. (2) ensures that the known data is preserved at locations in K, while the second term imposes a smoothness constraint on the missing pixels. For our experiments, we use a conjugate gradient scheme to determine the solution $\boldsymbol{u}$ of the linear system Eq. (2).

3. CODING IMAGES WITH SIMPLE INGREDIENTS

In the following, we first describe two of our core novelties, the joint inpainting and prediction (JIP) and a fast tonal optimisation with EFAN. In Section 3.3 we combine those components to our novel codec.

3.1. Joint inpainting and prediction

So far, inpainting-based codecs separate image reconstruction and efficient storage by prediction and/or entropy coding (e.g. PAQ [10]) completely. Instead, we propose to use EFAN from Section 2.1 for prediction and image reconstruction. We assume that the pixel values of $\boldsymbol{f}$ have been uniformly quantised to the grey level range $\{0, ..., q-1\}$.

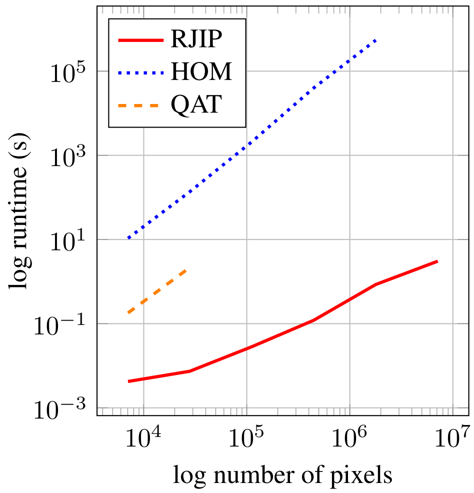
EFAN is implemented by visiting each known point $\boldsymbol{x}_j \in K$ sequentially and adding its contribution to the numerator of Eq. (1) to the value accumulation map $\boldsymbol{v}$ and its contribution to the denominator to the weight accumulation map $\boldsymbol{w}$. For $w_i := w(\boldsymbol{x}_i)$, this corresponds to the updates $w_i \leftarrow w_i + G(\boldsymbol{x}_i - \boldsymbol{x}_j)$ and $v_i \leftarrow v_i + G(\boldsymbol{x}_i - \boldsymbol{x}_j)$ for all locations $\boldsymbol{x}_i$ in the truncated Gaussian neighbourhood N_j of $\boldsymbol{x}_j \in K$. Finally, the new image is given by $u_i = v_i / w_i$.

Standard EFAN traverses all points of K before computing a reconstruction. However, as soon as the weight accumulation map w_i is non-zero at a location $\boldsymbol{x}_i$, a preliminary reconstruction $p_i = v_i / w_i$ can be defined. Even though this is only based on adjacent, already visited known data, it can act as a surprisingly accurate prediction.

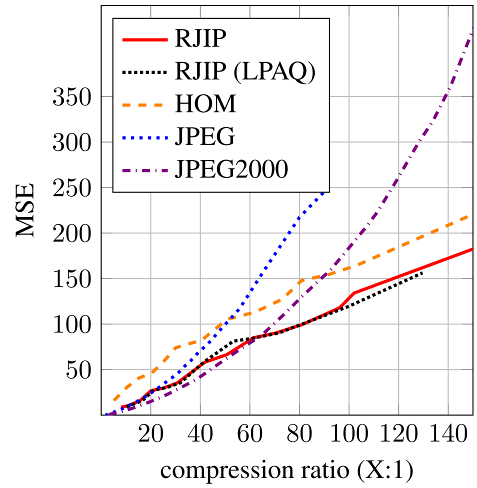
Therefore, our *joint inpainting and prediction* (JIP) extends EFAN by a simple prediction step: We visit all points $\boldsymbol{x}_i \in K$ in a fixed order (left to right, top to bottom) and estimate them by p_i if $w_i \neq 0$, and by 0 otherwise. By storing the prediction error $e_i = f_i - p_i \bmod q$ instead of the original pixel value f_i we can reduce the entropy of mask values. JIP requires only a single additional division for each prediction compared to standard EFAN. The adaptation of the truncated Gaussian's size to the mask density (see Section 2.1) ensures an overlap with at least one adjacent known data point that has not been visited so far. Thereby, starting with a single known data point, we can predict all remaining mask points.

(a) original image *trui*

(b) original image *sintel*



(c) runtime on *sintel*



(d) MSE on *trui*

Fig. 1. **(a)+(c) Runtime tests on *sintel*** show that RJIP outperforms HOM and QAT significantly. **(b)+(d) Qualitative tests on *trui*.** For cartoon-like images RJIP outperforms HOM and can also beat JPEG and JPEG2000.

For decoding, we perform the same JIP and load the inpainting errors e_i from the compressed image file. This allows us to recover the original data $f_i = p_i + e_i \bmod q$.

3.2. Fast tonal optimisation

Many inpainting-based codecs (e.g. [1, 2, 6, 16]) successfully use *tonal optimisation* to improve inpainting results. For given mask locations K, they do not store the original grey values $\boldsymbol{f}$, but instead choose pixel values that minimise the mean-squared error (MSE). Schmaltz et al. [2] visit all mask pixels in random order and change them to the next higher or lower quantisation if this decreases the MSE. Since each such check requires a full inpainting, this is a slow process.

For our EFAN-based codec, we can exploit the locality of the accumulation maps to speed up tonal optimisation significantly. Let u_i^{old} and u_i^{new} denote the original and new quantisation levels, and N_i the set of points in the truncated Gaussian neighbourhood of $\boldsymbol{x}_i$. The MSE change is then given by

$$\sum_{\boldsymbol{x}_j \in N_i} \left(f_j - \frac{v_j + G(\boldsymbol{x}_j - \boldsymbol{x}_i)(u_i^{\text{new}} - u_i^{\text{old}})}{w_j} \right)^2 . \qquad (3)$$

This updates the value accumulation map of each point $\boldsymbol{x}_j$ in the neighbourhood N_i: We substract the weighted old mask value from the map $\boldsymbol{v}$ and add the weighted new mask value. This localisation yields a significant speed-up.

3.3. The RJIP codec

We combine our new approach from the previous sections into a new ***r**egular grid codec with **j**oint **i**npainting and **p**rediction*: RJIP. Encoding consists of three steps.

Step 1: Mask selection and quantisation: Instead of intricate strategies to determine the mask location K, we place our known data on a regular grid with spacing h. This reduces reconstruction quality, but also removes the cost of storing positions. We also quantise the grey values uniformly to the range $\{0, ..., q-1\}$. The file header contains the image dimensions m, n (2–4 byte), h (1 byte), and q (1 byte).

Step 2: Fast tonal optimisation: We iterate our strategy from Section 3.2 until the MSE decreases by less than 0.001.

Step 3: Prediction and entropy coding: JIP from Section 3.1 yields a prediction error for the optimised grey values from Step 2. We encode these errors with finite state entropy (FSE) [13] and append the binary stream to the file header.

Since EFAN does not require to specify parameters, we only have to select the grid size h and the quantisation levels q. We use a binary search that minimises the MSE and ignores results that do not fulfil the desired compression ratio. For decoding, we extract the prediction errors from the file with FSE and apply JIP as described in Section 3.1.

4. EXPERIMENTS

In the following, we evaluate our RJIP codec w.r.t. speed and quality. For comparison to classic inpainting-based compression, we replace EFAN by homogeneous diffusion, yielding the codec HOM. JIP and fast tonal optimisation are not applicable for HOM. We also compare to JPEG (GraphicsMagick 1.3.21) and JPEG2000 (Kakadu 7.10.2).

JPEG	HOM	JPEG 2000	**RJIP**

MSE=160.57, SSIM=0.738	MSE=127.38, SSIM=0.751	MSE=109.906, SSIM=0.809	MSE=90.01, SSIM=0.827

Fig. 2. **Compression of *trui* at Ratio** $\approx 70:1$**.** RJIP outperforms HOM and JPEG visually and quantitatively. Compared to JPEG2000 the image is smoother, but does not suffer from artefacts while preserving more details in the hat.

4.1. Timing results

We investigate the scaling behaviour of RJIP and HOM on five downsampled versions of frame 917 of the 4K CinemaScope movie *sintel* (4096×1744, Fig. 1 (b)). All experiments are conducted on a single core of an Intel Core i7-6700A@3.40GHz with 32 GB RAM.

For a single compression with fixed parameters h and q, Fig. 1 (c) shows that RJIP outperforms HOM by up to five orders of magnitude. Its compression speeds range from 0.004s for a 128×55 image to 3.05s for a 4K image. Both algorithms have a similar memory requirement (≈ 380 MB for 4K). The main bottleneck for HOM is the tonal optimisation, since homogeneous diffusion cannot be localised as in RJIP. A faster approach for HOM, the quantisation-aware tonal optimisation (QAT) [16] trades speed for extensive memory consumption. Our test setup ran out of RAM for resolutions of 512×218 upwards. Even this complex algorithm is outperformed by RJIP by 2 to 3 orders of magnitude.

For decompression, HOM requires 4.80s for the full 4K resolution, while RJIP finishes in 0.39s. Overall, our new codec outperforms a comparable approach with homogeneous diffusion significantly in spite of the additional prediction.

4.2. Qualitative results

Due to space restrictions, our qualitative evaluation shows mainly MSE results. The perceptual structural similarity index (SSIM) [21] produced similar rankings (see also Fig. 2). Compression quality is not the main focus of our work, but we still aim to preserve the main benefits of inpainting-based compression: Good performance on cartoon-like images and high compression ratios.

The test image *trui* is a typical representative for such images. As Fig. 1(d) demonstrates, RJIP outperforms the slower HOM algorithm consistently over all compression ratios. The simpler EFAN inpainting not only approximates the more sophisticated homogeneous diffusion inpainting adequately, it

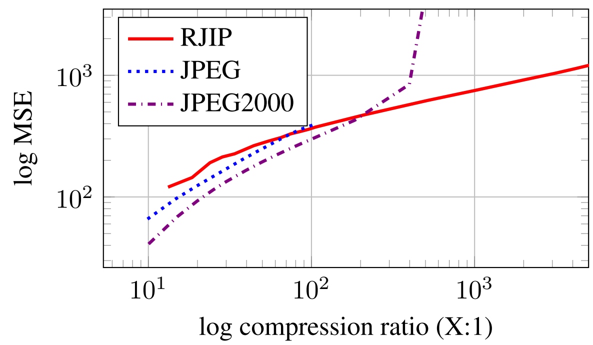


Fig. 3. **MSE on Berkeley database.** On the Berkeley database with textured images, RJIP outperforms JPEG and JPEG2000 for high compression ratios.

also benefits from the additional reduction of entropy by the new joint inpainting and prediction. This is supported by the fact that replacing JIP by the neural network predictions of LPAQ [10] yields no significant benefits. RJIP also outperforms JPEG for ratios $>$ 30:1 and JPEG 2000 for ratios $>$ 60:1. Visually, RJIP clearly outperforms HOM and JPEG in Fig. 2: It does not suffer from singularities as homogeneous diffusion inpainting, and has no block artefacts like JPEG. Compared to JPEG2000, RJIP results are slightly smoother but also have no wavelet quantisation artefacts.

In addition, we have also investigated the quality of RJIP on textured data, namely on the 500 natural images of the Berkeley database [22] in Fig. 3. As expected, JPEG and JPEG2000 perform more favourably compared to RJIP than on cartoon-like images. However, the simplicity of RJIP also leads to a significantly better scaling behaviour for ratios $>$ 100:1. Extreme compression ratios in the up to 5000:1 are possible.

5. CONCLUSION AND FUTURE WORK

Our RJIP codec combines simple ingredients to achieve faster and more accessible inpainting-based image compression. The combination of Shepard interpolation, joint inpainting and prediction (JIP), and fast global tonal optimisation is both easy to implement and outperforms conventional diffusion-based approaches by several orders of magnitude w.r.t. speed.

In particular, the concepts of localised inpainting and JIP have the potential for a high impact on the design of inpainting-based codecs. In particular, they seem well-suited for time-critical applications such as video coding [9]. Furthermore, the ease with which RJIP can obtain extreme compression ratios suggests that some of its concepts might be useful for specialised tasks like thumbnail compression [23].

In the near future, we plan to complement our fast tonal optimisation with a suitable time-efficient spatial optimisation to further improve compression quality. Moreover, we will address a dedicated support for colour images.

6. REFERENCES

[1] I. Galić, J. Weickert, M. Welk, A. Bruhn, A. Belyaev, and H.-P. Seidel, "Image compression with anisotropic diffusion," *Journal of Mathematical Imaging and Vision*, vol. 31, no. 2–3, pp. 255–269, July 2008.

[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, July 2014.

[3] W. B. Pennebaker and J. L. Mitchell, *JPEG: Still Image Data Compression Standard*, Springer, New York, 1992.

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

[5] 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, Sept. 2011.

[6] S. Hoffmann, M. Mainberger, J. Weickert, and M. Puhl, "Compression of depth maps with segment-based homogeneous diffusion," in *Scale-Space and Variational Methods in Computer Vision*, A. Kuijper, K. Bredies, T. Pock, and H. Bischof, Eds., vol. 7893 of *Lecture Notes in Computer Science*, pp. 319–330. Springer, Berlin, 2013.

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

[8] 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.

[9] S. Andris, P. Peter, and J. Weickert, "A proof-of-concept framework for PDE-based video compression," in *Proc. 32nd Picture Coding Symposium (PCS 2016)*, Nuremberg, Germany, Dec. 2016, pp. 1–5.

[10] M. Mahoney, "Adaptive weighing of context models for lossless data compression," Tech. Rep. CS-2005-16, Florida Institute of Technology, Melbourne, FL, Dec. 2005.

[11] D. Shepard, "A two-dimensional interpolation function for irregularly-spaced data," in *Proc. 23rd ACM National Conference*, Las Vegas, NV, Aug. 1968, pp. 517–524.

[12] R. Achanta, N. Arvanitopoulos, and S. Süsstrunk, "Extreme image completion," in *Proc. 42nd IEEE International Conference on Acoustics, Speech and Signal Processing*, New Orleans, LA, Mar. 2017, pp. 1333–1337.

[13] Y. Collet, "Finite state entropy (FSE) implementation," https://github.com/Cyan4973/FiniteStateEntropy, 2014, Last checked: January 23, 2019.

[14] J. Duda, "Asymmetric numeral systems: entropy coding combining speed of Huffman coding with compression rate of arithmetic coding," *arXiv 1311.2540*, Nov. 2013.

[15] H. Knutsson and C.-F. Westin, "Normalized and differential convolution," in *Proc. 1993 IEEE Conference on Computer Vision and Pattern Recognition*, New York, NY, USA, June 1993, pp. 515–523.

[16] P. Peter, S. Hoffmann, F. Nedwed, L. Hoeltgen, and J. Weickert, "Evaluating the true potential of diffusion-based inpainting in a compression context," *Signal Processing: Image Communication*, vol. 46, pp. 40–53, Aug. 2016.

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

[18] U. Y. Desai, M. M. Mizuki, I. Masaki, and B. K. P. Horn, "Edge and mean based image compression," Tech. Rep. 1584 (A.I. Memo), Artificial Intelligence Lab, Massachusetts Institute of Technology, Cambridge, MA, Nov. 1996.

[19] J. Gautier, O. Le Meur, and C. Guillemot, "Efficient depth map compression based on lossless edge coding and diffusion," in *Proc. 29th Picture Coding Symposium*, Kraków, Poland, May 2012, pp. 81–84.

[20] T. Iijima, "Basic theory of pattern observation," in *Papers of Technical Group on Automata and Automatic Control*, Kyoto, Japan, 1959, IECE, In Japanese.

[21] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[22] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. Eigth International Conference on Computer Vision*, Vancouver, Canada, July 2001, pp. 416–423.

[23] D. Marwood, P. Massimino, M. Covell, and S. Baluja, "Representing images in 200 bytes: Compression via triangulation," in *Proc. 25th IEEE International Conference on Image Processing (ICIP)*, Athens, Greece, Oct. 2018, pp. 405–409.