MULTI-MODE INTRA PREDICTION FOR LEARNING-BASED IMAGE COMPRESSION

Henrique Costa Jung[⋆]*, Nilson Donizete Guerin Jr.*[⋆]*, Raphael Soares Ramos*[⋆]*, Bruno Macchiavello*[⋆]
Eduardo Peixoto[⋆]*, Edson Mintsu Hung*[⋆]*, Teofilo de Campos*[⋆]
Renam Castro da Silva[†]*, Vanessa Testoni*[†]*, Pedro Garcia Freitas*[†]

[⋆]Universidade de Brasília [†]Samsung R&D Institute Brazil

ABSTRACT

In recent years image compression techniques based on deep learning have achieved great success and their performances are gradually reaching the methods crafted by experts, such as JPEG, WebP, and Better Portable Graphics (BPG). A technique that is fundamental for modern image and video codecs is intra prediction, which takes advantage of local redundancy to predict the pixels from previously encoded neighbors. In this paper, we use Convolutional Neural Networks (CNN) to develop a new intra-picture prediction mode. More specifically, we propose a multi-mode intra prediction approach that uses two CNN-based prediction modes and all intra modes previously implemented in the High Efficiency Video Coding (HEVC) standard. We also propose a bit allocation technique that increases the bitstream only if the reconstruction error is significantly reduced. Experimental results evince a significant and consistent performance increase compared to other approaches that use a similar backbone architecture, with a 28% bitrate reduction compared to the baseline codec.

***Index Terms*—** image codec, intra prediction, CNN autoencoder, variable rate compression

1. INTRODUCTION

Image coding techniques exploit spatial or spectral redundancies to decrease the number of bits required to represent the images. These techniques include prediction-based coding, transform-based coding, subband coding, block truncation coding, vector quantization, etc. [1]. Although early image coding standards, such as the JPEG standard [2] and later the JPEG 2000 standard [3], are based on image transforms, most of the image coding standards are based on hybrid schemes that use a combination of more than one of these techniques. Among the possible hybrid schemes, the combination of transform and predictive coding is presumably the most successful. In fact, this combination is widely adopted in state-of-the-art image codecs such as BPG [4], WebP [5], and JPEG-XL [6] as well as to encode the intra frames in most video coding standards [7, 8, 9, 10]. The basic idea behind prediction-based coding schemes is to predict the pixel values of a given image block based on the previously encoded neighborhood. Then only the residual, that is the difference between the original and predicted blocks, is encoded. It is expected that the residual information will have a smaller dynamic range and variance compared to the original image, which can benefit the compression algorithms.

Recently, with the rapid development of deep learning theory, learning-based approaches have been proposed as an alternative to traditional coding schemes. Ballé et al. [11, 12] proposed some of the most pertinent methods in the area. In [11], they introduce a rate-distortion control during training and an estimation of the entropy of the encoded data to enhance the CNN architecture used to compress data. Although this method achieves competing coding performance, a clear drawback of this architecture is that it needs to be specifically trained for each bitrate. On the other hand, Toderici et al. [13] proposed one of the first architectures for encoding images using autoencoders that presents a distinguishing characteristic of being scalable and not requiring multiple training to achieve the desired bitrate. Toderici's architecture can control quality and rate according to the number of iterations, meaning that one decoder fits all possible qualities. Toderici's work was then improved by Minnen et al. [14], which extended the model to use intra prediction before encoding.

In this paper, we improve upon Toderici's [13, 14] proposal. Similarly to the Advanced Video Coding (AVC) and HEVC codecs, our proposed method includes a multi-mode intra prediction, with a block-based choice of the best mode. AI prediction modes have a very good performance on a large number of scenarios, but on simpler patches, traditional intra modes are better. We also propose a simple but efficient bit allocation strategy that decides the number of iterations at the block level instead of the whole image.

2. BACKGROUND

One of the approaches to end-to-end image compression is based on a cascade of neural networks, specifically, a cascade of autoencoders [13]. Each autoencoder compresses its input and tries to reconstruct it. The residual error is propagated through the next autoencoder. Considering E as the encoder,

Part of the results presented in this work was obtained through the *Deep Codec* project funded by Samsung Eletrônica da Amazônia Ltda under the Brazillian Informatics Law 8.248/91. TdC thanks CNPq PQ 314154/2018-3. BM thanks CNPq PQ 308548/2018-3.

D as the decoder, B a binarization function, and x the input (raw data), an autoencoder can be represented as:

$$F(x) = D(B(E(x))) \ . \tag{1}$$

This equation can be used to compose a cascade of residual autoencoders by the following set of equations:

$$\begin{aligned} r_0 &= x \\ F_t(r_{t-1}) &= D_t(B(E_t(r_{t-1}))) \\ r_t &= r_{t-1} - F_t(r_{t-1}) \ , \end{aligned} \tag{2}$$

where $F_t(r_t)$ represents the reconstruction of the data r_t at the t-th iteration. The pair (E_t, D_t) is the autoencoder for which r_{t-1} is the input. The binarizer B, in this formulation, is the same for all iteration levels.

In the first proposed model using this approach [13], each network is composed of an encoder, which produces a latent representation of a fixed number of bits and a decoder which tries to reconstruct the input image from the latent. In the same study, the use of recurrent layers is introduced and better exploited later [15]. In another work, intra prediction was introduced [14], where each block is predicted using a separate CNN that receives neighboring reconstructed blocks as input and a blacked-out region that represents the block that is currently being encoded (see Fig. 1(b)). Then the prediction is subtracted from the raw block to produce the residual information which is given as input to the autoencoder.

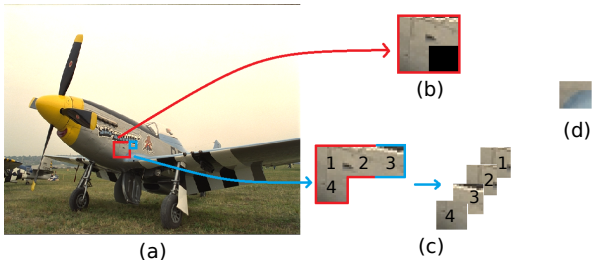


Fig. 1. Example of an input given to the CNN intra prediction modes: (a) original image, (b) input for the first CNN intra mode, (c) input for the second CNN intra mode, (d) target.

3. PROPOSED ARCHITECTURE

In conventional image codecs, like BPG [4], intra prediction is a very efficient technique, since it usually yields a Laplacian distributed residual data that normally has lower entropy compared to the original data [16]. In [14] a single intra prediction mode, based on Machine Learning (ML) inpainting, was introduced for the iterative autoencoder approach.

Conventional image encoders have several intra prediction modes to allow for the selection of the best mode in terms of rate-distortion. In order to build an intra prediction candidate, usually, the pixels at the left and top borders are extrapolated to the current block in a certain direction. Several modes are tested, and the best candidate is used at the encoder. In the HEVC, 35 intra prediction modes are available, being one planar mode, one DC mode, and 33 directional modes [10].

Our proposed architecture includes all 35 HEVC intra prediction modes in the iterative autoencoder using recurrent layers. Moreover, two intra modes created using CNNs are introduced. The first mode was developed by Minnen et al. [14], while the second one is proposed in this paper. Hence, our encoder has 37 possible modes to be selected for intra prediction. Once the best prediction x' is selected, the input for the first iteration is modified to $r_0 = x - x'$, then the autoencoder works as described by eq. (2). This process is depicted in Fig. 2.

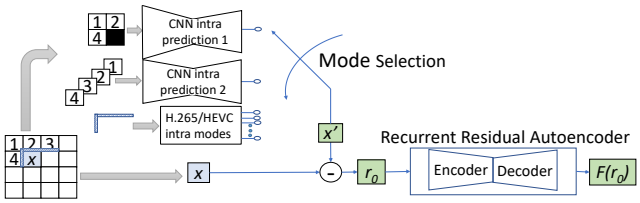


Fig. 2. Block diagram of the proposed method. The residual autoencoder can be run for multiple iterations, depending on the reconstruction quality.

The HEVC modes can produce an accurate prediction in several regions of the image, particularly where there is a directional texture pattern. On the other hand, the CNN modes can generate much richer images, because of their non-linear capabilities, including predictions of curves and more general shapes.

3.1. CNN-based Intra Prediction

In this section, we detail the two CNN intra prediction modes used in our codec: the first is based on the previous proposal by Minnen et al. [14], and the second novel proposal presented here. The other 35 intra prediction modes used in our codec are a standard implementation of the HEVC intra prediction modes and are detailed elsewhere [10].

3.1.1. Minnen's Prediction Mode

In this mode, the CNN input is an image block of 64×64 pixels, with the bottom right quadrant blacked out, as depicted in Fig. 1-(b). In the training stage, the original bottom-right sub-image is given to the network as the target, as shown in Fig. 1-(d). This is consistent with a raster-scan order prediction, except for blocks on the first row and first column (missing neighbor blocks are considered to have pixel values 128). The network has a 64×64 image with 3-color channels input, we pass it through 4 convolutional layers with kernels of size 4×4, and zero padding. Then, we pass it through a depthwise convolution layer, followed by a reshaping and a pointwise convolution layer, which is also known as separable convolution. Then again, we pass it through 3 transposed

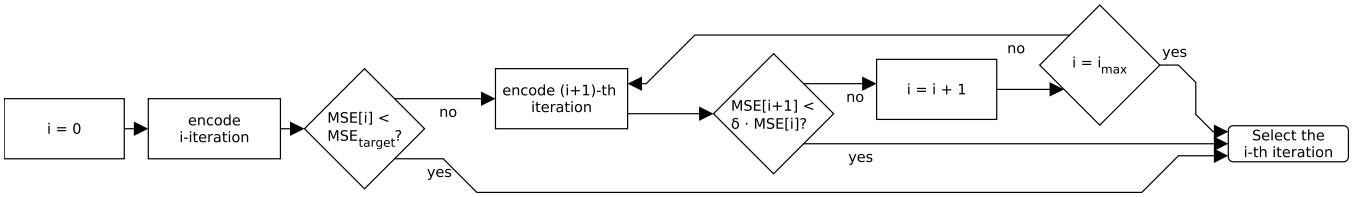


Fig. 3. Flowchart of the proposed Bit allocation strategy. In the figure, $\mathrm{MSE}\,[i]$ is the MSE for the i-th iteration.

convolution layers, generating an output image of 32×32 pixels. Between each convolutional layer, we pass the filter responses through a Rectified Linear Unit (ReLU) activation function.

3.1.2. Proposed CNN-based intra prediction mode

A drawback of the approach described in Section 3.1.1 is that the black pixels are part of network input. It increases the complexity of the training procedure because the first layers will have to learn to ignore the sudden drop of the input signal and the edge that this introduces to the input images. Furthermore, there is no reason to omit the upper right diagonal neighbor block from the input. This block would already have been coded and may provide valuable information for the network. Hence, the proposed CNN-based intra prediction mode receives four 32×32 blocks neighboring the current block (in this order): upper left diagonal, top, upper right diagonal and left, as shown in Fig. 1-(c). The blocks are concatenated in an additional dimension. Therefore, we changed the first layers to deal with this additional dimension: the first layer is a 3D convolutional layer with kernel $4 \times 4 \times 4$, and the second layer is also a 3D convolutional layer with kernel $2 \times 4 \times 4$.

3.2. Proposed Residual Autoencoder

Using all the 37 possible prediction modes when encoding a block, the encoder selects the best mode, i.e. the mode that creates a better reconstruction in terms of Mean Squared Error (MSE). This comes at a cost of a single parameter in the bitstream of each block signaling the chosen mode – in our implementation, this is simply encoded with a 6-bit word. The architecture of the residual autoencoder is detailed in Table 1. The same network was used for our experiments with and without intra prediction.

3.3. Bit Allocation

In the original architecture of the iterative model [13] all blocks use the same bitrate (i.e., the same number of iterations) regardless of the quality achieved for each block. In another work, the authors presented a bit allocation technique for this architecture [17], where each block would use a different number of iterations (i.e., different bitrate) depending on how the quality of that block approaches a given target quality. In our codec, the use of intra prediction makes this even more important – the prediction for some blocks is very close to the block itself so that it is not needed to spend more bits for that block. To properly address this issue, we propose a Bit Allocation (BA) technique that is simpler than that of [17] and, more importantly, it is applied only during the tests – there is no need to perform a specific training or to modify a previously trained model to use it.

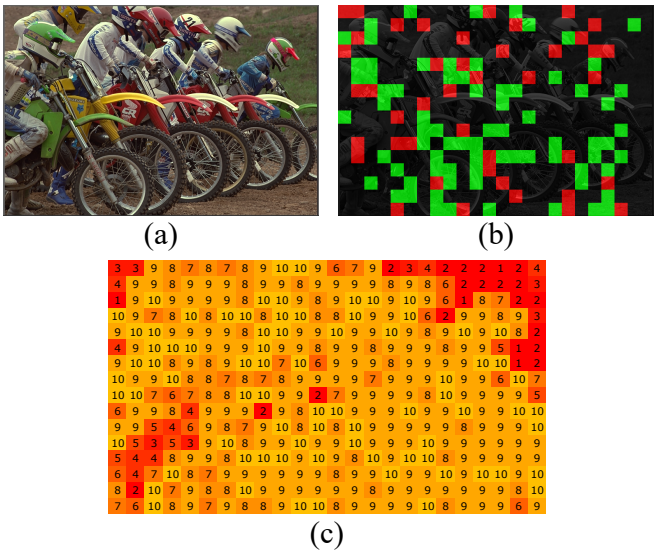
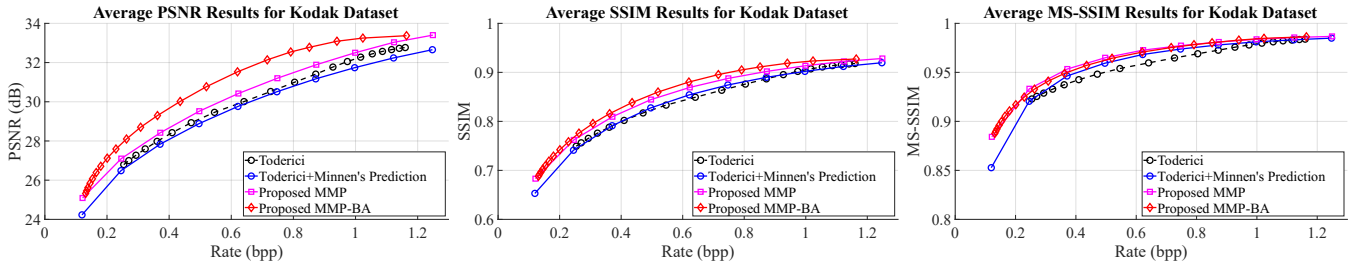


Fig. 4. Example of Prediction and Bit Allocation: (a) Original Image; (b) Prediction Map (the color represents which mode was chosen per block: grey, red and green stand for HEVC Modes, base CNN and proposed CNN, respectively); and (c) Number of Levels (i.e., the number of iterations chosen by the bit allocation algorithm).

The iterative encoding occurs on a block basis, at each iteration, the encoder evaluates a few conditions to decide whether it is worth spending more bits. As shown in Fig. 3, at each iteration, if the achieved MSE is lower than the MSE_{target}, the encoder stops encoding and proceeds to the next block. If the achieved MSE is higher, the encoder performs another encoding iteration and check $\mathrm{MSE}\,[i+1] < \delta \times \mathrm{MSE}\,[i]$. If the MSE is not sufficiently lower, the encoder stops encoding, otherwise, it proceeds to the next encoding iteration. The encoding stops when one of these conditions is true or the maximum number of iterations is achieved.



(a) PSNR (b) SSIM (c) MS-SSIM

Fig. 5. Average results of the Proposed Codec using (a) PSNR, (b) SSIM, and (c) MS-SSIM as metrics for the Kodak dataset.

Table 1. Architecture of the Residual Recursive Autoencoder. All convolutional layers use zero padding, except for the one just before the binarizer. The spatial resolution of all convolutional filters is 3×3, except for the ones just before and after the Binarizer, which are 1×1. The LSTM layers use only 1 hidden layer. This architecture has a total of 32,835,360 parameters.

	Layer	Output Shape	Strides	Activation
Encoder	Input Layer	$32 \times 32 \times 3$	-	-
	Conv	$16 \times 16 \times 64$	2	Linear
	ConvLSTM	$8 \times 8 \times 256$	2	Sigmoid
	ConvLSTM	$4 \times 4 \times 512$	2	Sigmoid
	ConvLSTM	$2 \times 2 \times 512$	2	Sigmoid
	Conv	$2 \times 2 \times 32$	1	Tanh
	Binarizer	$2 \times 2 \times 32$	-	-
Decoder	Conv	$2 \times 2 \times 512$	1	Linear
	ConvLSTM	$2 \times 2 \times 512$	1	Sigmoid
	DepthToSpace	$4 \times 4 \times 128$	-	-
	ConvLSTM	$4 \times 4 \times 512$	1	Sigmoid
	DepthToSpace	$8 \times 8 \times 128$	-	-
	ConvLSTM	$8 \times 8 \times 256$	2	Sigmoid
	DepthToSpace	$16 \times 16 \times 64$	-	-
	ConvLSTM	$16 \times 16 \times 128$	2	Sigmoid
	DepthToSpace	$32 \times 32 \times 32$	-	-
	Conv	$32 \times 32 \times 3$	1	Tanh

4. RESULTS AND DISCUSSION

The residual autoencoder was trained using the CLIC [18] database for $300,000$ iterations. For a fair comparison, all CNN models use the same input database and the same training hyper-parameters (batch size, learning rate, etc.). We use as anchor the base architecture from [13], without intra prediction and bit allocation. All tests were performed on the Kodak dataset [19].

Fig. 4 illustrates the chosen modes for one test image. We can notice that HEVC modes are chosen for most of the patches. This is expected since it represents 35 different modes, and directional interpolation is good enough for most parts in a regular image. However, the CNN-based modes are selected a significant amount of times, which means that the HEVC modes are not able to generate a good enough prediction in certain cases. Considering the whole test dataset, we observe that mode selection rates are 19.2% for Minnen's prediction mode, 19.6% for the proposed CNN-based mode, and 61.2% for HEVC prediction modes. Additionally, we have also noted that the CNN prediction modes are selected more often at low bit rates images, and in high entropy blocks.

Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM) [20], and Multi Scale SSIM (MS-SSIM) [21] quality scores are depicted in Fig. 5. This figure shows the rate-distortion curve for Toderici's [13], Minnen's [14], the proposed Multi-Mode Prediction (MMP) architecture, and the Multi-Mode Prediction with Bit Allocation (MMP-BA). From these curves, we can notice that Minnen outperforms the Toderici in SSIM and MS-SSIM, but not in PSNR. The proposed MMP approach surpasses them all metrics, and the proposed MMP-BA has even better PSNR and SSIM results.

We computed the bitrate savings as the average BD-Rate [22] savings per image in the Kodak Dataset, using Toderici's baseline codec as the anchor. Using single-mode Minnen's prediction, the average bitrate increase was $+11.7\%$, ranging from $+59\%$ to -14%, i.e., using a single prediction actually performs worse for most images. Note that single-mode will use intra prediction even if it decreases compression efficiency. Using MMP, the average savings was 5.9%, ranging from $+17\%$ to -25%. Finally, using MMP-BA, the average savings was -28.1%, ranging from $+4\%$ to -50%. Using MMP, each patch needs 0.15s to be compressed, and 0.19s for MMP-BA. All models occupy 5.7G on a GPU.

5. CONCLUSIONS

In this paper, we introduce the HEVC intra modes in an iterative image autoencoder in order to create a multi-mode intra prediction approach. We also include two CNN-based intra prediction modes, one of them based on a previous proposal and another that is an improvement upon that mode. By comparing the codecs with and without the proposed intra prediction, we can see a consistent performance gain when using the HEVC and the CNN-based prediction modes. We also propose a bit allocation technique that only increases the bitstream if the reconstruction error is significantly reduced and can be used with no CNN retraining. As future work, the use of intra prediction can be applied in other architectures and the use of attention models can be applied to bit allocation.

6. REFERENCES

[1] Majid Rabbani and Paul W Jones, *Digital image compression techniques*, vol. 7, SPIE press, 1991.

[2] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Transactions on Consumer Electronics*, vol. 38, pp. 18 – 34, Feb. 1992.

[3] D. Taubman and M. Marcellin, "JPEG2000: standard for interactive imaging," *Proceedings of the IEEE*, vol. 90, pp. 1336 – 1357, Aug. 2002.

[4] Fabrice Bellard, "The BPG image format," Online, April 21 2018, Available from `http://bellard.org/bpg/`.

[5] Richard Rabbat, "WebP a new image format for the web," Chromium Blog. Google, September 30 2010, `https://blog.chromium.org/2010/09/webp-new-image-format-for-web.html`.

[6] Jyrki Alakuijala, Ruud van Asseldonk, Sami Boukortt, Martin Bruse, Zoltan Szabadka, Iulia-Maria Coma, Moritz Firsching, Thomas Fischbacher, Evgenii Kliuchnikov, Sebastian Gomez, et al., "JPEG XL next-generation image compression architecture and coding tools," in *Applications of Digital Image Processing XLII*. International Society for Optics and Photonics, 2019, vol. 11137, p. 111370K.

[7] ITU-T, "ITU-T Recommendation H.263, Video coding for low bit rate communication," Tech. Rep., ITU-T, February 1998.

[8] ITU-T, "ITU-T Recommendation H.264, advanced video coding for generic audiovisual services," Tech. Rep., ITU-T, March 2005.

[9] R. L. de Queiroz, R. S. Ortis, A. Zaghetto, and T. A. Fonseca, "Fringe benefits of the H.264/AVC," in *International Telecommunications Symposium*, Sep. 2006, pp. 166–170.

[10] ITU-T, "ITU-T Recommendation H.265, high efficiency video coding," Tech. Rep., ITU-T, April 2013.

[11] Johannes Ballé, Valero Laparra, and Eero P. Simoncelli, "End-to-end optimized image compression," in *5th International Conference on Learning Representations (ICLR)*, 2017, Preprint available at `http://arxiv.org/abs/1611.01704`.

[12] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston, "Variational image compression with a scale hyperprior," in *International Conference on Learning Representations (ICLR)*, 2018, Preprint available at `http://arxiv.org/abs/1802.01436`.

[13] G. Toderici, S. M. O'Malley, S. J. Hwang, D. Vincent, D. Minnen, S. Baluja, M. Covell, and R. Sukthankar, "Variable rate image compression with recurrent neural networks," in *International Conference on Learning Representations (ICLR)*, Apr. 2016.

[14] David Minnen, George Toderici, Michele Covell, Troy Chinen, Nick Johnston, Joel Shor, Sung Jin Hwang, Damien Vincent, and Saurabh Singh, "Spatially adaptive image compression using a tiled deep network," in *International Conference on Image Processing (ICIP)*, Feb. 2017.

[15] George Toderici, Damien Vincent, Nick Johnston, Sung Jin Hwang, David Minnen, Joel Shor, and Michele Covell, "Full resolution image compression with recurrent neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5306–5314.

[16] Khalid Sayood, *Introduction to Data Compression*, Elsevier - Morgan Kaufmann, 2012.

[17] Michele Covell, Nick Johnston, David Minnen, Sung Jin Hwang, Joel Shor, Saurabh Singh, Damien Vincent, and George Toderici, "Target-quality image compression with recurrent, convolutional neural networks," *ArXiv*, vol. abs/1705.06687, 2017.

[18] George Toderici, Michelle Covel, Wenzhe Shi, Radu Timofte, Licas Theis, and Johannes Ballé, "Dataset of the challenge on learned image compression (CLIC)," 2019, Available from `http://www.compression.cc/2019/challenge/`.

[19] Rich W. Franzen, "Kodak image dataset," Online, Updated on January 27 2013, Available from `http://r0k.us/graphics/kodak/`.

[20] 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, April 2004.

[21] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers*, May 2003.

[22] Gisle Bjøntegaard, "Improvements of the BD-PSNR model," Tech. Rep., VCEG-AI11, ITU-T SG16/Q6, Berlin, Germany, July 2008.