https://doi.org/10.2352/ISSN.2470-1173.2021.10.IPAS-259

HEVC Rate-Distortion Optimization with Source Modeling

Ahmed M. Hamza; University of Portsmouth; Portsmouth, Hampshire, UK
Mohamed Abdelazim; University of Portsmouth; Portsmouth, Hampshire, UK
Abdelrahman Abdelazim; Blackpool and the Fylde College; Blackpool, UK
Djamel Ait-Boudaoud; University of Portsmouth; Portsmouth, Hampshire, UK

Abstract

The Rate-Distortion adaptive mechanisms of MPEG-HEVC (High Efficiency Video Coding) and its derivatives are an incremental improvement in the software reference encoder, providing a selective Lagrangian parameter choice which varies by encoding mode (intra or inter) and picture reference level. Since this weighting factor (and the balanced cost functions it impacts) are crucial to the RD optimization process, affecting several encoder decisions and both coding efficiency and quality of the encoded stream, we investigate an improvement by modern reinforcement learning methods. We develop a neural-based agent that learns a real-valued control policy to maximize rate savings by input signal pattern, mapping pixel intensity values from the picture at the coding tree unit level, to the appropriate weighting-parameter. Our testing on reference software yields improvements for coding efficiency performance across different video sequences, in multiple classes of video.

Introduction

Several factors in design allow the x265/HEVC (High Efficiency Video Coder) to have the large coding efficiency gain (signal data compression) over the prior x264 video codec. As a hybrid video encoding system, a mixture of modes are adapted as needed, with greater granularity/flexibility in decisions than previous standards in both intra and inter modes. Intra, for encoding based on spatial info redundancy in the same frame, has more directional modes than prior codecs, and inter-coding (temporal multi-picture referencing) has a multitude of new tools, including a weighted merge-mode mechanism.

To improve in sequences of large picture size (Full-HD, Ultra-HD 4k video and beyond), a larger basic block size (CTU, or Coding Tree Unit) is employed, and a multitude of new division Modes in the comprised CUs (Coding Units) and the associated, but independently sized, Prediction and Transform Units (PU and TU) for the Coding Units. Changes to the encoding decision process ultimately affect the referencing candidate choices and the chosen mode/depth for encoding each CU, before the slices of CTU are transformed[1], quantized and coded per the standard (see [2]).

Notably, and of interest to us here, the decision making process[3] in the Rate-Distortion algorithms of the encoder are based on more elaborate Lagrangian-weighted cost functions than before, with a finer-grained control of the Lagrangian λ that is not solely based on Quantization Parameter (QP) value – the original approach in [4]. We apply here recent advances from visual Reinforcement Learning algorithms to the problem, using a simple neural architecture for function estimation and control. The goal is to further optimize the weighted cost functions for improved compression (efficiency) in all encoding modes and frame referencing levels.

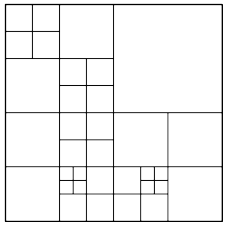


Figure 1: An example of CTU mode decision (block splitting) flexibility in the current HEVC/x265 scheme.

Rate-Distortion Optimization in H.265/HEVC

Various referential candidate choices in both temporal and spatial redundancy modes, determine the ultimate mode decisions and overall prediction structure of every independent coding block in MPEG codecs (x263, x264, x265); here we focus on the function particulars of HEVC/x265 in recent versions [5], [6].

At the heart of the Rate-Distortion Optimization procedures of the encoder, several cost functions are used to determine best candidate picks, refinement of motion estimation (inter mode), or other courses of action (CU splitting depth or shape). Our focus is on the cost functions J_{SSE} and J_{SAD}:

$$J_{SAD} = D_{SAD}(s,c) + \lambda_{MOTION} \cdot R \tag{1}$$

where D_{SAD} is mean absolute error (distortion), and R is the rate. A similar D_{SSE} exists for mode decisions, using the sum-of-squared-error functions, which requires the λ_{SSE} version of the Lagrangian to be $\lambda_{MODE} = (\lambda_{SAD})^2$.

The minimization of the cost is a goal of the RDO algorithms and candidate searches, while taking care of burst/volatile rate behavior, and the rate-constrained or QP-oriented quantization further downstream. The current encoder versions derive a well-performing Lagrangian λ_{MODE}, by frame, using QP and frame-level information from a lookup table [7].

Prior to this, the value was simply related to quantization as a negative slope of distortion-rate function, for a particular QP:

$$\lambda_{MODE} = -\frac{\delta D}{\delta R} \tag{2}$$

This was approximated in circa-2001 encoders (see [4], [8]) as $0.85 \times Q^2$, for a different set of quantization tools than those in use today. Current encoder adaptive methods (see reference encoder

documentation [5]) use a more fine grained adaptation, still QP-derived and weighted:

$$\lambda_{MODE} = \alpha \cdot W_k \cdot 2^{(QP-12)/3.0} \tag{3}$$

Here the adaptive factors α and W_k are further granular measures of adaptation, assuming a constant source distribution, and that the relation holds for CUs in a frame overall. This granularity is at the frame level.

We investigate in this work the addition of source information and automated control of weighting by a control agent that *learns* the modeling aproximation by trial, error and reward (Reinfocement).

Methods and Algorithms

In this section we present our proposed contribution, which is an application of several reinforcement learning ideas (exploration and control) with adaptation to our problem setting.

Preliminaries

Reinforcement Learning, broadly speaking, involves processes of optimizing the behavior of an agent/controller in an environment, through interaction with the environment. This is typically formulated as being time-sequential state space, consisting of environment states $x \in X$, action space $u \in U$, the notion of a policy of action $\pi_\theta : x \mapsto u$, to be learned, which controls the agent in an optimal manner through reward-maximizing mechanism..

Unlike *supervised* learning scenarios, where correct labels are *known* for respective state inputs, the process by which a control agent learns in the reinforcement setting is based on the indications of cost, which is a quantitative sum of action rewards $(r_1, r_2, \ldots)$ governed by some system of reward assignment related to the feedback from the environment. This may be episodic, and chosen with respect to the algorithm or environment.

Reward can be assigned in a greedy manner (immediate observation per step), or through multiple experience roll-outs (i.e., a history) culminating towards some final state, of the *observe* $\rightarrow$ *action* $\rightarrow$ *reward*, there may be a γ-discounted reward, or a simple average of per-timestep reward assigned per the full result of the rollout episode.

A cost function for the policy can be written as:

$$J(\pi_\theta) = E_\tau[\sum_{i \in S} r(x_i, u_i)] \tag{4}$$

The above Eq.(4) represents the expected sum of rewards, and is to be maximized by the training procedure. This can be stated in recursive Q-learning formulation, in terms of the current reward and all rewards thereafter with the same policy:

$$\begin{aligned} Q^{\pi_\theta}(x_t, u_t) = & \mathbb{E}[r(x_t, u_t)] \quad + \\ & \mathbb{E}[max(Q(x_{next}, u_{next}))] \end{aligned} \tag{5}$$

which is learned by updating network parameters θ through backpropagation on reward signals. The updates to the policy are a conditioning on the environment response to the outputs of the network.

Policy Gradient methods (see [9], [10] are characterized as actor-only methods, and are simpler in one aspect due to ability to converge on a real-valued policy through gradient descent/ascent, so that an optimal cost (or maximal reward is found). The gradient of this cost function is:

$$\nabla_\vartheta J = \frac{\partial J}{\partial \pi} \frac{\partial \pi}{\partial \vartheta} \tag{6}$$

which is estimated per time-step update, so that the parameters ϑ (e.g: neural network signal weights) can be updated via some learning rate α:

$$\vartheta_{knext} = \vartheta_k + \alpha \nabla_\vartheta J_k \tag{7}$$

This overall approach very closely maps to our situation in the control of video encoding parameters – in the current problem setting, we have no optimal or known labels to train an agent with in a supervised manner, because we do not (in this work) view the reference implementation values for the RDO hyper-parameters as ground truth labels. That is, supervised learning methods are unnatural to the problem.

Instead, we aim to discover an approximation of the unknown ideal *for each portion of a given sequence*, by modeling, exploration, and overall reward assignment.

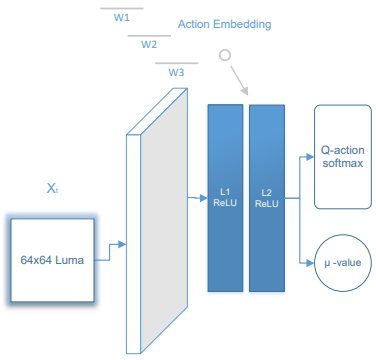


Figure 2: Function Approximator NN for the learning procedure. The relationships between source signal, current encoding level state and action-result are captured by the stacked low dimensional layers with full Regularized Linear connections.

Reinforcement Learning for Encoder Control

A computationally efficient, pixel-value-based algorithm giving real-valued (continuous, non-discrete) function estimates is desired. The main algorithms we are utilizing in this paper are based on Continuous versions of Q-learning by Gu et al.[11], which shares a neural network architecture and purpose with [12], and a similar continuous control method, with backpropagation updates to a neural net function approximator in [13]. This is in turn based on earlier work on policy gradients for continuous-valued actions in [14].

Our method is a combination of actor-only (policy gradient; continuous value output) and Q-learning (discrete value prediction for action choice), which yields desirable properties for an encoder control agent that is to augment encoder decisions at runtime.

This critic learns discrete responses of action categories per input luminescence pattern, including *none*, to keep the same lambda unchanged. Q-function critic outputs a selection of 3 possibilities

(lower, higher and constant) for the λ weighting, given source signal and current picture reference level, and an exploration policy of randomized action that is judged by bit-rate cost as reward.

Since it is difficult to arrive at precise, real-valued actions (weights) with just the critic network, the Q-function is approximated by a neural net that is also shared by a similarly parameterized actor policy unit. The actor, or discrete policy gradient, is guided by episodic reward signals to converge on values for the lambda weighting for coding efficiency gain.

Other properties include *robustness* in the method as a whole; stability in changing source signal environments, general convergence and possible transfer-ability of learned policy behavior.

Our application to HEVC/H.265 involves two distinct procedural stages: First, the modeling function is trained with roll-out episodes of encoding that incorporate randomized exploration, and train the function (neural net) parameters.

Recall the common formulation for Q-function above in Preliminaries (Eq. 5). To indicate expectation of reward and state from history rollout, we write this as:

$$Q^{\pi_\theta}(h_t,u_t) = \mathbb{E}_{h_t}[r(x_t,u_t) \quad + \\ max(Q(h_{next},u_{next}))] \quad (8)$$

The exploration factor (during training) is a randomized variation added to the Q-function action choice:

$$u_t = \mu^\theta(x_t) + \aleph \quad (9)$$

So that the action is based on the parametrized (neural approximated) function of the input signal, plus noise. Training is done by incorporating reward feedback as parameter updates upon taking this action (encoding the CTU) and taking further actions (encoding the slice). The procedure is presented below, in *Training*.

Adaptations

Some specifics in procedure arise from our encoding optimization setting. One major point is that, unlike supervised learning methods, many techniques are excluded from the RL algorithms (large networks, batch normalization, etc). Second, the notion of state transition does not exist for our case. We rely purely on reward observation that is devoid of resulting next-state, because the reward is directly obtained (continuous) and the next image is inconsequential to the goal (we do not observe the reward from the same state-space at the next time step).

In addition, there is no goal state. So, in the critic Q-function formula, the policy π_θ can be updated by a summation over $Q(x_t,\pi_\theta(s_t,\zeta_t))$ where ζ_t comes from the transition model: $s_{t+1} = f(s_t,a_t) + \zeta_t$. Our method simply measures reward values and discards this term, the dynamics of which need not be learned.

The reward signal is based on the RD measured values for the *reference* encoding of the same unit. The final reward for a roll-out (a slice encoding, with all comprosed CTUs) is the improvement in bit-rate cost of that slice as a whole, with no discounting factor for time. The reward per discrete *time-step* is the same, but per CTU.

Training Algorithms

Our version of the deterministic Policy Gradient Procedure for learning is summarized as follows:

Algorithm 1 Policy Gradient Exploration for RDO-Improvement

procedure TRAIN(Q_θ)
  Initialize randomized policy π_θ
  for slice-encoding trials SL_1, SL_2 **do**
    for $CTU_1, CTU_2, \cdots \in$ SLICE **do**
      Act (Encode) and store result ▷ RD
      update action-value network π_θ
      update Q_θ per instant result
    end for
    Update both networks cumulatively
by rollout result
  end for
  return π, Q
end procedure

An estimate of bit-rate reduction per *lambda* parameter variation is the goal of our critic portion, through exploration and parameter update. Upon training on one or more sequences, the controller-actor is ready to be deployed in the process of encoding the video sequence using the inferential output, without further exploration. This is an off-line procedural design.

The process is "model–directed", in that the inferred values are *weight-modified* values for the Lagrangian λ_{MODE}, and associated λ_{ME}, based on the initial empirical estimate in the HM lookup-table of values. Therefore our method can be thought of as an exploration and refinement, mapping input signals to proposed shifts in the weighting values for different CTU settings.

Keeping in mind the ultimate target for the reward mechanism is an encoder decision choice leading to either: 1) an improved (reduced) bitrate cost estimate (without penalty) or 2) an improved distortion level at the same bitrate cost level (i.e., without penalty).

In practice, most decision change in the encoder will lead to at least slight penalty in one side of the Lagrangian function, depending on the source image structure. However, just as the fixed values chosen by lookup-table in HM reference are considered empirically better in most scenarios *on average* (all possible sequences and input images), our source-adapted behavior can sometimes lead to a significant jump in reward, in some blocks, by crossing a minor decision threshold, as shown in the *Results* section.

Algorithm 2 The weight update inference/filter function.

function MODIFYW(a)
  $D \leftarrow$ INVOKE-CRITIC(QP)
  if $D = 0$ **then**
    return a ▷ Built-in HM value
  end if
  return UPDATEW(a, D) ▷ Policy network value
end function
Require: $0.5 \leq a \leq 40.0$

Encoding Results and Analysis

Results of experimentation with the HM14.5 reference encoder are presented below.

Figure 3: (Final, learned) Lambda Weighting Factor variation in the first 100 frames of encoding. The variation was clipped (as in HM) to provide stability in RD performance and prevent over-compensation in distortion penalties.

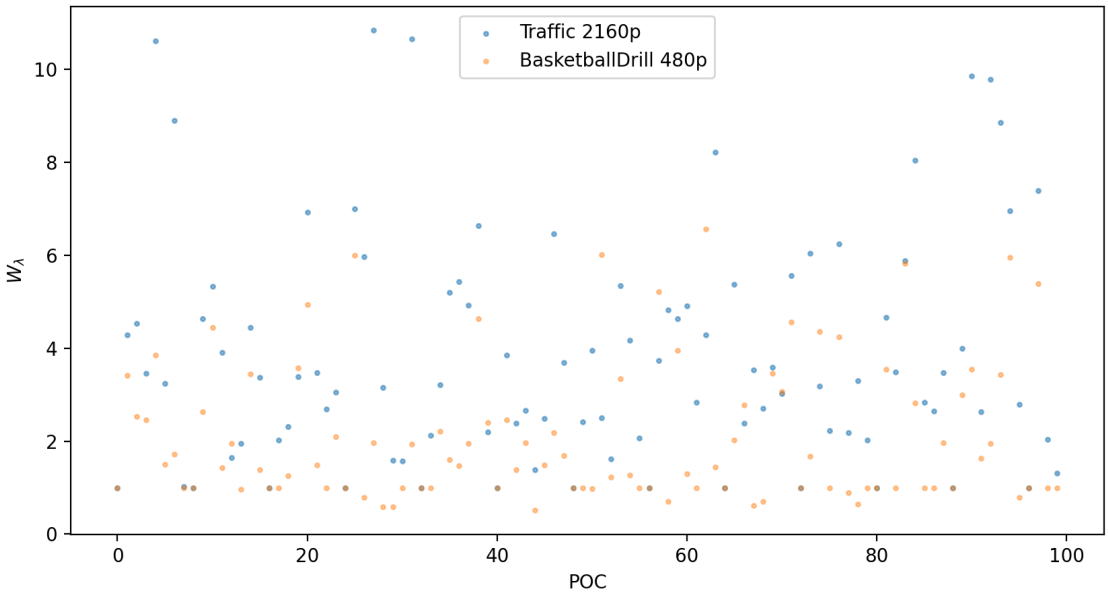
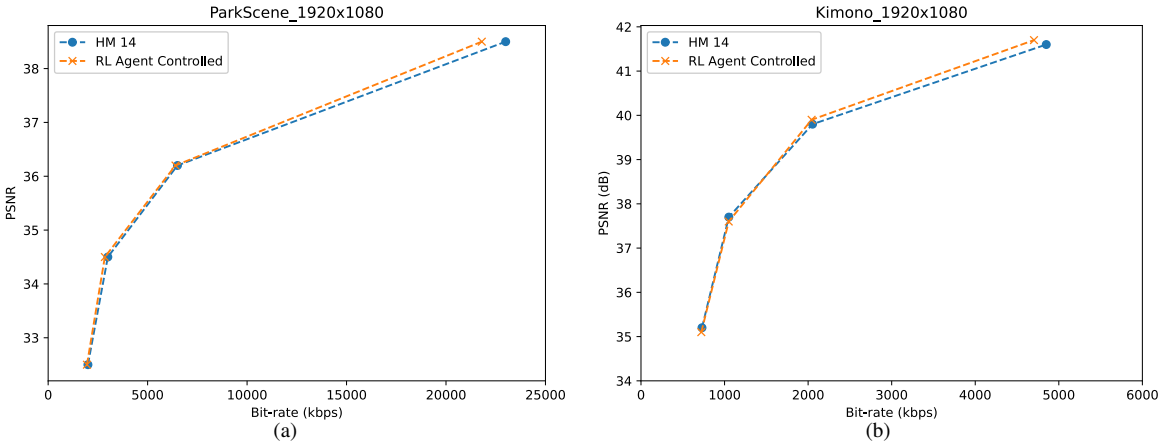


Figure 4: Rate-Distortion curves for several classes 4a and of test video sequence. The overall effect of the agent control of lambda is evident at higher resolution frame samples.



Our training design is based on the reference HM software (see [7]) and its Python-wrapped data structures in [15], for easy prototyping. The results we obtained during testing/execution phase were almost identical in VideoLAN's free implementation of HEVC encoder software, which we adopted for part of the testing due to being optimized for speed (and multiple times as fast).

The network architecture adapted from [11], [16] is shown in Fig. 2. We used no convolutional layers (no filters needed), and introduced actions embedded in the second hidden layer. Low-dimensional hidden layers were both ReLU based, and fully connected with 200 units.

Several test sequences (per JCT-VC guidelines[17]) were tested, and show that a trained agent can provide significant improvements to BD-bitrate performance as a whole, to the full sequence encoding when run during testing. This holds true for different *types* of video content, especially when SKIP modes are not predominantly benefited from.

When trained with a reward function that kept bitrate as a goal while also providing positive reinforcement (at a lower reward factor) to distortion reduction, the results were nearly identical in most sequences, but allowed a full improvement in both PSNR-Y (Peak Signal to Noise Ratio, Luma-based) and bit-rate (fully quantized/encoded sample size) simultaneously, raising the rate-distortion curves in some frames. Note that this improvement is *parametrized on input* and not a fixed choice per sequence.

The training-testing procedures were carried out using the same sequences, with testing repeated for each QP level: $QP \in \{22, 27, 32, 37\}$. Table 1 shows Bjorntegard-Delta average curve differences for both rate and distortion.

The mode decisions in Fig. 5 below are visualized [1] to show regions affected by the agent lambda choice. We find that in pre-

[1] We used the open-source GITL x265 Visualizer available in [18]

Table 1: Average Rate-Distortion performance of our learned policy λ vs. the HM reference λ.

Seq. Name	Fr. Encoded	BD-PSNR-Y (dB)	BD-R (%)
Traffic	150	-0.078	-3.2
FourPeople	600	0.061	-4.1
BasketballDrill	300	-0.710	-2.1
Kimono	240	-0.090	-3.7
ParkScene	300	0.070	-4.2
Cactus	500	0.021	-6.5
Tennis	600	-0.015	-5.3
Vidyo4	600	-0.141	-2.5
Average	N/A	-0.311	3.95

Table 2: Testing sequences info.

Class	Resolution	Name	Frame-rate
A	2560×1600	Traffic	30
B	1920×1080	Kimono	24
B	1920×1080	Cactus	50
B	1920×1080	ParkScene	24
C	832×480	BasketBallDrill	30
E	1280×720	FourPeople	60
E	1280×720	Vidyo4	60

liminary comparison, few of the CTUs in still or slow moving regions are affected in terms of coding efficiency, because the encoder is already intelligently applying a SKIP signal to those blocks, which in place regardless of our modifications to the cost function outcomes. Therefore, the slice regions benefitting the most from the learned adaptive λ_{π^ϑ}, are the areas that are non-skipped in the reference (i.e., motion estimation required), particularly in large CU modes (32-pixel).

Algorithmic Complexity

Here we present the asymptotic complexity of the algorithms and the empirical observations, to better comprehend the practicality of the results aside from their empirical coding efficiency benefits. To better gauge how such an approach would work with higher-classed video sequences (Level 5-6 resolutions and above), scalable sequences, HEVC-3D extensions, 10 and 12 bit sequences, 8K video and so forth. Computing power on mobile devices is ever increasing, but a fair estimate of asymptotic performance will show the applicability to more demanding video encoding scenarios regardless of the rate of increase of hardware performance.

The current two-step process of training before real-time execution requires a linear increase in computation time that is dependent on sequence length and picture sizes. Training happens over rollouts comprising smaller episodes, where the rollouts are slice-oriented (so in the 100-600 count range) and the CTU-based *obervation* $\rightarrow$ *action* cycle is replayed 5 times per episode. We used the entire frame sequence for training, since simulations in RL typically involve 10^4 to 10^7 iterations, depending on replay memory, algorithm and environment. This means larger picture resolutions underdo more training (e.g., 1080p containing more CTUs than 420p).

Activation passes (action selection and policy value estimation) are feed-forward and constant time per CTU decision. Overall, our full setup (with training phase) involved an encoding time penalty equivalent to an additional run of the encoder with QP-delta variation settings activated.

Related Work

Several works in both reinforcement learning (see [19]–[21]) and video coding optimization have arisen in recent years, building on the successes of basic machine intelligence architecture, particularly deep convolutional and recurrent (memory-baed) neural nets, in visually-oriented tasks. Of relevance to our work, we note the content-based QP adaptations in Intra-type coding[22], which is incorporated in HEVC reference tools, specifically for intra. Some recent work aiming at complexity reduction by intelligent prediction in the encoder rate-distortion processes has also been done in [23], and more intra-mode complexity reduction efforts can be seen in [24], [25] and [26]. In [27], several architectures including Long-Short-Term-Memory neural nets are utilized within the encoder, again for complexity reduction by learning to predict the CU division mode and structure.

Conclusions

Our work shows the ability of combined reinforcement agents (Q-learning and policy gradient) to learn refinements of encoding parameter control by input image patterns as state information. This is a novel addition to MPEG-HEVC optimization, and the first application of policy-learning agents (as opposed to direct classification schemes) to video coder control.

Several improvements are possible in our process, particularly a move to online learning, avoiding a two-step process that was used in our initial work for clarity and verification of performance. Also, we have not established the transfer of agent capability by training on several sequences at a time – this is notably important, in that it allows a ready-made function approximator NN to be produced. Are the agent's performances on one set of videos transferable with similar parameter weights to other sequences? And how quickly can convergence occur on unseen sequences? Finally, there is a possibility of including sequential state information, via memory-keeping networks (LSTM), e.g., recurrent memory preserving systems in [12], [19]. This has direct application to Low-delay profile settings in MPEG-HEVC and similar codecs.

References

[1] T. Nguyen, P. Helle, M. Winken, B. Bross, D. Marpe, H. Schwarz, and T. Wiegand, "Transform coding techniques in HEVC," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 6, pp. 978–989, Dec. 2013. DOI: 10.1109/JSTSP.2013.2278071.

[2] G. J. Sullivan, J. R. Ohm, W. J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012, ISSN: 1051-8215. DOI: 10.1109/TCSVT.2012.2221191.

[3] J. L. Lin, Y. W. Chen, Y. W. Huang, and S. M. Lei, "Motion vector coding in the HEVC standard," *IEEE Journal of Selected Topics*

in Signal Processing, vol. 7, no. 6, pp. 957–968, Dec. 2013, ISSN: 1932-4553. DOI: 10.1109/JSTSP.2013.2271975.

[4] T. Wiegand and B. Girod, "Lagrange multiplier selection in hybrid video coder control," in *Proceedings 2001 International Conference on Image Processing (Cat. No.01CH37205)*, vol. 3, Oct. 2001, pp. 542–545. DOI: 10.1109/ICIP.2001.958171.

[5] C. Rosewarne, B. Bross, M. Naccari, *et al.*, "High efficiency video coding (HEVC) test model 16 (HM16) Improved Encoder reference software, JCTVC-W1002," JCT-VC, JCT-VC Output Document, 2016, https://phenix.int-evry.fr/jct/doc_end_user/current_document.php?id=10479.

[6] JCTVC, *High Efficiency Video Coding test model 15 (HM15) reference software, JCTVC-Q1002*, 2014.

[7] JCT-VC, *HEVC reference software 16.0 [ONLINE]*, https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/.

[8] T. Wiegand, M. Lightstone, D. Mukherjee, T. G. Campbell, and S. K. Mitra, "Rate-distortion optimized mode selection for very low bit rate video coding and the emerging h.263 standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 2, pp. 182–190, Apr. 1996, ISSN: 1051-8215. DOI: 10.1109/76.488825.

[9] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, Nov. 2017. DOI: 10.1109/MSP.2017.2743240.

[10] I. Grondman, L. Busoniu, G. A. D. Lopes, and R. Babuska, "A survey of actor-critic reinforcement learning: Standard and natural policy gradients," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 1291–1307, 2012.

[11] S. Gu, T. Lillicrap, I. Sutskever, and S. Levine, "Continuous deep q-learning with model-based acceleration," in *Proceedings of The 33rd International Conference on Machine Learning*, M. F. Balcan and K. Q. Weinberger, Eds., ser. Proceedings of Machine Learning Research, vol. 48, New York, New York, USA: PMLR, Jun. 2016, pp. 2829–2838. [Online]. Available: http://proceedings.mlr.press/v48/gu16.html.

[12] N. Heess, J. J. Hunt, T. P. Lillicrap, and D. Silver, "Memory-based control with recurrent neural networks," in *NIPS Workshop on Deep Reinforcement Learning*, 2015. [Online]. Available: http://rll.berkeley.edu/deeprlworkshop/papers/rdpg.pdf.

[13] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," in *Proc. International Conference on Learning Representations (ICLR)*, https://ui.adsabs.harvard.edu/abs/2015arXiv150902971L, May 2016.

[14] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ser. ICML'14, Beijing, China: JMLR.org, 2014, pp. 387–395.

[15] D. Springer, W. Schnurrer, A. Weinlich, A. Heindel, J. Seiler, and A. Kaup, "Open Source HEVC Analyzer for Rapid Prototyping (HARP)," in *IEEE Int. Conf. on Image Processing (ICIP)*, Paris, France, Oct. 2014.

[16] B. O'Donoghue, R. Munos, K. Kavukcuoglu, and V. Mnih, "Combining policy gradient and q-learning," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017. [Online]. Available: https://openreview.net/forum?id=B1kJ6H9ex.

[17] F. Bossen, *Common hm test conditions and software reference configurations*, http://phenix.it-sudparis.eu/jct/doc_end_user/current_document.php?id=7281, Group: MPEG-H, Mar. 2013.

[18] H. Li. (2016), [Online]. Available: https://github.com/lheric/GitlHEVCAnalyzer.

[19] M. Hausknecht and P. Stone, "Deep recurrent q-learning for partially observable mdps," in *AAAI Fall Symposium on Sequential Decision Making for Intelligent Agents (AAAI-SDMIA15)*, Arlington, Virginia, Nov. 2015.

[20] V. Mnih, N. Heess, A. Graves, and k. kavukcuoglu koray, "Recurrent models of visual attention," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., Curran Associates, Inc., 2014, pp. 2204–2212. [Online]. Available: http://papers.nips.cc/paper/5542-recurrent-models-of-visual-attention.pdf.

[21] M. P. Deisenroth, G. Neumann, and J. Peters, "A survey on policy search for robotics," *Found. Trends Robotics*, vol. 2, no. 1–2, pp. 1–142, Aug. 2013, ISSN: 1935-8253. DOI: 10.1561/2300000021. [Online]. Available: http://dx.doi.org/10.1561/2300000021.

[22] A. Zhang and D. Bull, "HEVC enhancement using content-based local qp selection," English, in *2016 IEEE International Conference on Image Process (ICIP 2016)*, ser. Proceedings of the IEEE International Conference on Image Processing (ICIP), IEEE ICIP 2016 ; Conference date: 25-09-2016 Through 28-09-2016, United States: Institute of Electrical and Electronics Engineers (IEEE), Mar. 2017, pp. 4215–4219, ISBN: 9781467399623. DOI: 10.1109/ICIP.2016.7533154.

[23] T. Nguyen Canh, M. Xu, and B. Jeon, "Rate-distortion optimized quantization: A deep learning approach," in *IEEE High Performance Extreme Computing Conference*, Sep. 2018.

[24] P. Hensman, "Intra-prediction for video coding with neural networks," M.S. thesis, KTH ROYAL INSTITUTE OF TECHNOLOGY, Mar. 2018.

[25] R. Birman, Y. Segal, A. David-Malka, and O. Hadar, "Intra prediction with deep learning," vol. 10752, 2018. DOI: 10.1117/12.2320551. [Online]. Available: https://doi.org/10.1117/12.2320551.

[26] J. Li, B. Li, J. Xu, R. Xiong, and W. Gao, "Fully connected network-based intra prediction for image coding," *IEEE Transactions on Image Processing*, vol. 27, no. 7, pp. 3236–3247, Jul. 2018, ISSN: 1057-7149. DOI: 10.1109/TIP.2018.2817044.

[27] M. Xu, T. Li, Z. Wang, X. Deng, R. Yang, and Z. Guan, "Reducing complexity of hevc: A deep learning approach," *IEEE Transactions on Image Processing*, vol. 27, no. 10, pp. 5044–5059, Oct. 2018, ISSN: 1057-7149. DOI: 10.1109/TIP.2018.2847035.

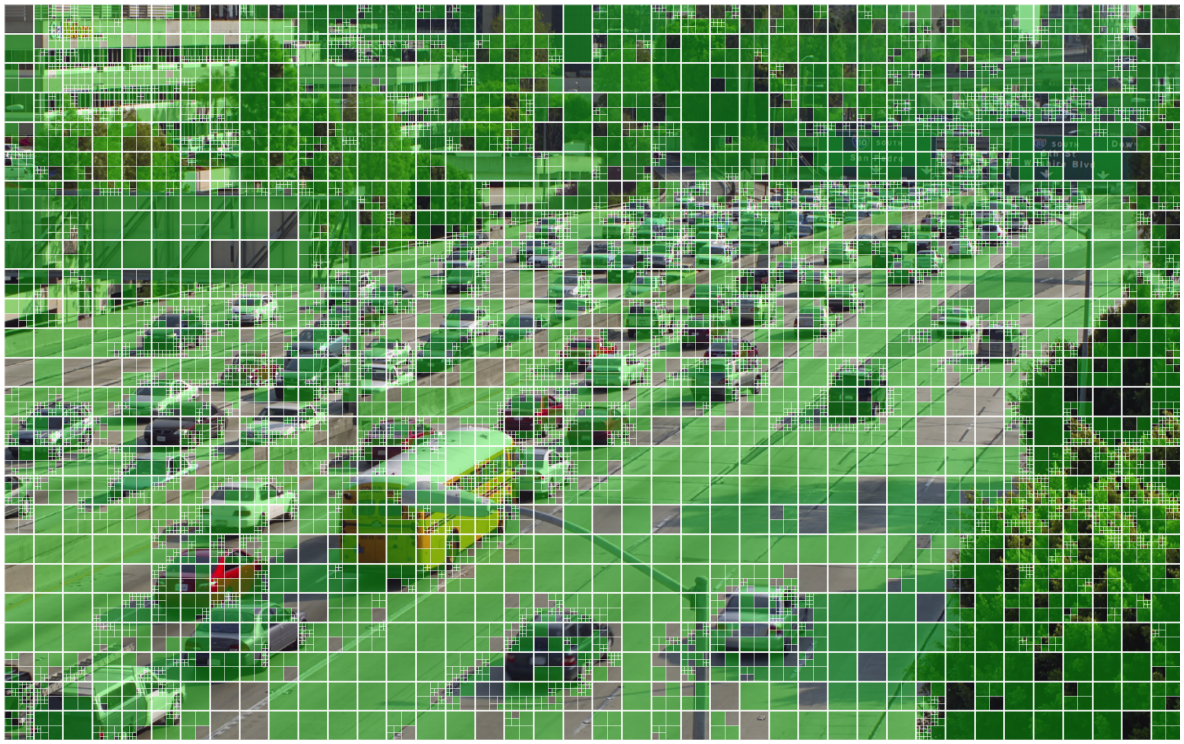
Author Biography

Ahmed M. Hamza is a PhD candidate in the School of Computing at the university of Portsmouth, UK, and teaches Computing and Cybersecurity at RIT, in New York. His research focus is in reinforcement learning and machine intelligence applications to software control and video coders. He obtained his M.S in Computer Science 2010 from Georgetown University, with a thesis on applied algorithms in chem-informatics.

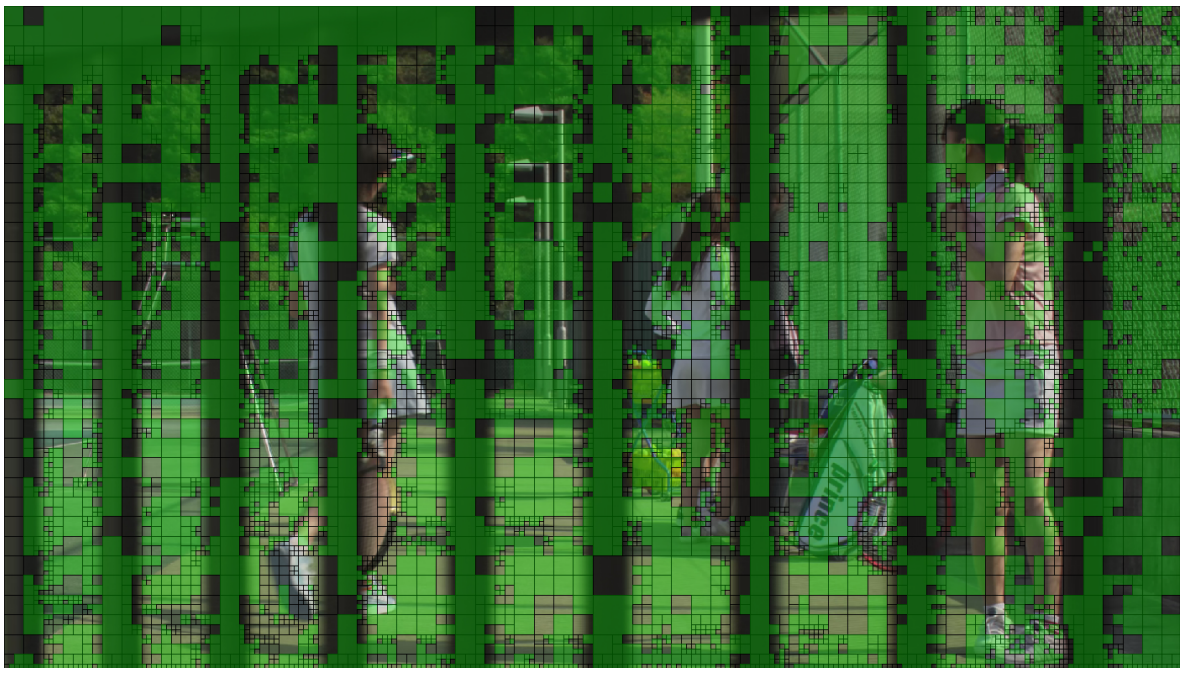
Mohamed Abdelazim received his BS in computer engineering from Cairo University 2005. Mohamed worked in Mentor Graphics (A Siemens business) from 2013 to 2019. He is currently working as software engineer in Amazon Canada, and a PhD student at University of Portsmouth. His work in Mentor Graphics and Amazon focuses on componential geometrics algorithms. His PhD research domain is using machine learning in video encoding and super resolution.

Abdelrahman Abdelazim is an Engineering Curriculum Manager at Blackpool and the Fylde College, UK. He holds a BEng (Hons) degree in Digital Communication and a PhD degree in Engineering, both from the University of Central Lancashire (UCLAN), Preston, UK. Between 2008 and 2012 he worked as Lecturer in Electronics within the School of Computing, Engineering and Physical Sciences (CEPS). From 2012 to 2017 he was Associate Professor and Head of Department at the American University of the Middle East (AUM). His research interests are in the area of reducing the complexity of Video Coding Encoders in real-time Scalable and Multi-view applications.

Djamel Ait-Boudaoud joined the University of Portsmouth in 2010 and is currently Professor and Dean of the Faculty of Technology. Before Portsmouth he was the head of the School of Computing, Engineering and Physical Sciences at the University of Central Lancashire for close to 10 years. His research interests are predominantly focused on the problems of optimisation with applications in 3D computer vision, video standards (H264) and solving combinatorial (ordered sequence) problems using evolutionary algorithms. He gained a PhD in 1991 from the Department of Electrical and Electronic Engineering at University of Nottingham, UK., is a Chartered Engineer (CEng) and a fellow of the Institution of Engineering and Technology (FIET).



(a) Traffic (2560 × 1600)



(b) Tennis (1080p)

Figure 5: CTU mode decision (block splitting) visualization for the Traffic and ParkScene frames in inter-mode coding. The reference candidates in (b) are shown by the blue and red lines. Larger CUs within each CTU block can benefit from bitrate savings.

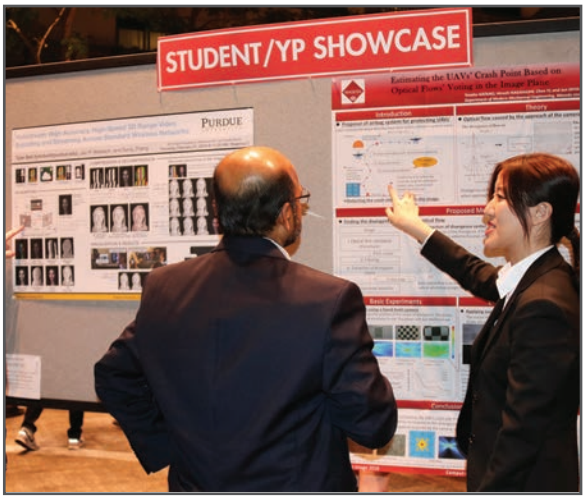
JOIN US AT THE NEXT EI!

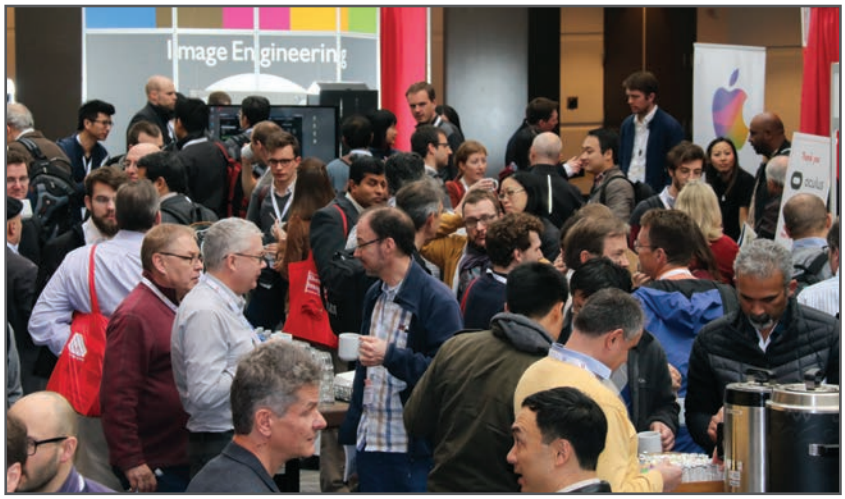
IS&T International Symposium on

Electronic Imaging

SCIENCE AND TECHNOLOGY

Imaging across applications . . . Where industry and academia meet!





• SHORT COURSES • EXHIBITS • DEMONSTRATION SESSION • PLENARY TALKS •
• INTERACTIVE PAPER SESSION • SPECIAL EVENTS • TECHNICAL SESSIONS •

www.electronicimaging.org

IS&T
imaging.org