A NEW ROTATION SEARCH FOR DEPENDENT RATE-DISTORTION OPTIMIZATION IN VIDEO CODING

Tuukka Toivonen, Loren Merritt**, Ville Ojansivu*, and Janne Heikkilä**

*Machine Vision Group
Infotech Oulu and Department of Electrical and Information Engineering
P. O. Box 4500, FIN-90014 University of Oulu, Finland
{tuukkat,vpo,jth}@ee.oulu.fi

**24-4 Appleway, Kalispell MT 59901 USA
Affiliation:
Department of Computer Science
University of Washington
lorenm@u.washington.edu

ABSTRACT

We present a novel search algorithm which is suitable for optimizing functions with a high-dimensional discrete-valued parameter vector. The algorithm is designed to find a function local optimum with the minimal number of evaluated points without requiring function derivatives. The algorithm is applied to frame-level rate-distortion (R-D) optimization using Lagrangian relaxation to the rate constraints and to block motion estimation in H.264-based video coding. The R-D optimization is further accelerated by finding a good starting point by the golden section search. The results show excellent near-optimal R-D performance while computation is reduced by 99% compared to the quadratic coordinate-wise steepest descent algorithm. In motion estimation, the new algorithm requires 7-13% less checking points than the small diamond search algorithm with only a small penalty in prediction quality.

Index Terms— Rate-distortion optimization, Motion estimation, Optimization methods

1. INTRODUCTION

The objective of rate-distortion (R-D) optimization in video coding is to select the operational parameters of the encoder to produce as good video quality as possible with the available bit budget [1]. There are many parameters that can be optimized, such as macroblock coding modes and transform coefficients. In this work we investigate mainly the selection of frame-level quantization parameters (QPs) in a H.264 encoder, but consider also motion estimation (ME).

Let N be the number of frames in a video sequence. For each frame n, a QP q_n selects the tradeoff between the bit rate R_n and the video distortion D_n. The total distortion of the entire sequence is $D_{\text{tot}} = \sum_{n=0}^{N-1} D_n$ and the total rate is $R_{\text{tot}} = \sum_{n=0}^{N-1} R_n$. All modern video encoders are predictive: a frame is predicted by using one or several previously encoded frames. Thus, if a previous frame is similar to the current frame and the previous frame is encoded with good quality, the current frame can be encoded with very small bit rate while still obtaining good quality (small distortion). Therefore, we can not assume that D_n and R_n are functions of q_n alone but functions of possibly all of the QPs for the whole sequence $\overline{Q} = \left[\begin{array}{ccc} q_0 & \cdots & q_{N-1} \end{array} \right]$.

The objective is to find QPs which minimize the distortion subject to the maximum bit rate: $\overline{Q}^* = \arg\min_{\overline{Q}} D_{\text{tot}}\left(\overline{Q}\right)$ subject to $R_{\text{tot}}\left(\overline{Q}\right) \leq R_{\max}$, where $\overline{Q}^* = \left[\begin{array}{ccc} q_0^* & \cdots & q_{N-1}^* \end{array} \right]$ is the optimum vector of QPs. The Lagrangian multiplier technique allows converting this constrained problem into unconstrained one by introducing the Lagrangian multiplier λ [1]:

$$C\left(\overline{Q}\right) = D_{\text{tot}}\left(\overline{Q}\right) + \lambda R_{\text{tot}}\left(\overline{Q}\right) \quad (1)$$

$$\overline{Q}^* = \arg\min_{\overline{Q}} C\left(\overline{Q}\right). \quad (2)$$

By choosing λ and minimizing Eq. (2), we obtain the optimal QP vector for the resulting rate. When only the rate is given, the value for λ can be found, for example, by bisection search, or by predicting it approximately from a previously encoded video sequence [1]. The remaining problem is to find the optimal QP vector $\overline{Q}^*$, which is the main subject of this paper.

An obvious method to find the discrete function optimum is exhaustive search: try all possible QP combinations and select the best. This is infeasible in practice, because the number of combinations is exponential in the number of frames, and evaluating the function for each combination is computationally very burdensome. In [2], the problem is formulated into a trellis search and the monotonicity assumption is used to prune as many branches from the trellis as possible. The

work assumes that a frame is predicted from at most two previously encoded frames (with B-frames), which is true for MPEG-2; however, in H.264 a frame may depend on several (up to 16) previously encoded frames. Also, a trellis search is intricate and has potentially exponential complexity. Thus, the quadratic coordinate-wise steepest descent (QCWSD) algorithm was presented in [3]. It is very simple and produces near-optimal result assuming the reachability property. Unfortunately the QCWSD algorithm is impractically slow for longer video sequences. Its speed was later improved by reducing the number of necessary derivative calculations [4]. Nevertheless, even the faster linear coordinate-wise steepest descent search (LCWSD) is slow, and furthermore, it often fails to find near-optimum QPs.

In this paper, we introduce a new search algorithm called the rotation search (RS), which is designed to find the function local optimum with the minimal number of function evaluations. As we do not make any assumptions about the function to be optimized, the algorithm can also be applied for other purposes. Although the RS is guaranteed to find only a local optimum, in practice it gives results very near to the globally optimal algorithms in finding the QPs.

Another problem in video coding is block ME. In this case, we are optimizing function similar to (1), but the distortion is computed using sum of absolute differences between the current macroblock and a candidate block in the search area, and the rate is obtained from a look-up table based on the motion vector length [5]. In ME, the function to be optimized has only two dimensions: the motion vector vertical and horizontal components. We apply the RS also to ME and find it to be faster than the small diamond search (SDS).

2. THE ROTATION SEARCH

The RS algorithm works better when a good initial position for the search can be predicted. Therefore, for the R-D optimization, we first initialize the QP vector using the one-dimensional golden section search (GSS) [6]. The GSS can be used to quickly find the minimum of the function by assuming that the QPs in all frames would be equal. This assumption converts the N-dimensional function parameter vector into one-dimensional. When the RS is used for ME, we use the median motion vector predictor [5], as defined in the standards, for the initial search location.

The RS algorithm (shown below) iteratively loops over all N dimensions. For each dimension, the direction, into which the function is minimized, is predicted from the previous loop execution for the same dimension. A step into this direction is then immediately taken by increasing or decreasing the QP by one. If the function value decreases, the prediction was correct, the step is taken, and the loop continues to the next dimension. If the function value instead increases, a step into the opposite direction is taken. If the function increases also into this direction, we are at a local minimum in this particular dimension. Otherwise, the prediction is updated so that at the next time the first step is taken into this same direction. As soon as a step is not taken into any direction in any dimension, the algorithm has found a local minimum. Since there always is a minimum, this guarantees that the algorithm finishes in a finite number of steps. We call the algorithm "rotation search", because it rotates over the dimensions as long as it finds the local optimum.

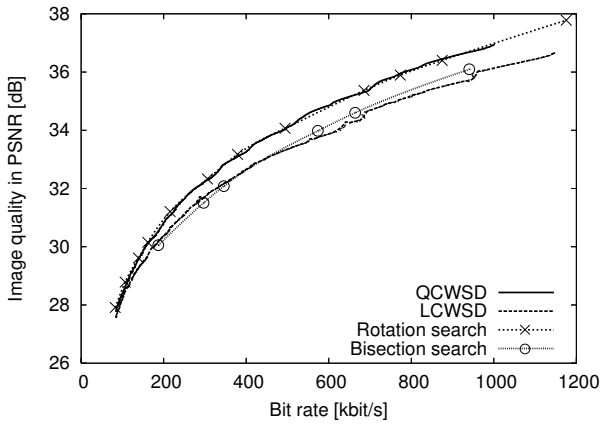
$\overline{U}_d$ is a vector of dimensionality N, with all other elements zero except the dimension $d \in \{0, \ldots, N-1\}$, which is 1. For example, $\overline{U}_2 = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}$ when $N = 4$.

$\overline{Q} :=$ initial guess from the GSS
$p_n := 1$ for $n \in \{0, \ldots, N-1\}$ # prediction
$c := 2N$ # downcounter
$d := 0$ # dimension
$i := 1$ # direction
loop
  if $C\left(\overline{Q} + \overline{U}_d \times i \times p_d\right) < C\left(\overline{Q}\right)$ **then**
    # New minimum: take the step, update prediction
    $\overline{Q} := \overline{Q} + \overline{U}_d \times i \times p_d$
    $p_d := p_d \times i$
    # Reset downcounter, skip to next dimension
    if $i = -1$ **then**
      $c := 2N$
    else
      $c := 2N + 1$
    end if
    $i := -1$
  end if
  $i := -i$
  if $i = 1$ **then**
    $d := (d + 1) \mod N$
  end if
  $c := c - 1$
while $c > 0$
return $\overline{Q}$

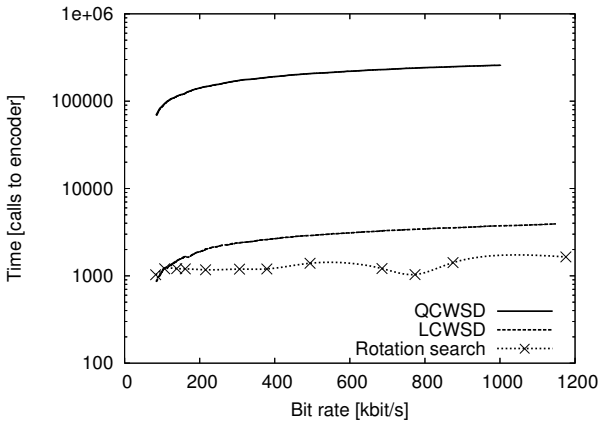
3. EXPERIMENTAL RESULTS

Five different R-D optimization algorithms were implemented in the C language: QCWSD, LCWSD, RS, bisection search, and exhaustive search[1]. The bisection search assumes that the QP is equal in all frames and performs an one-dimensional search. The exhaustive search tries all QP combinations with values between 27 and 45 ($19^5 = 2476099$ calls to encoder with five frames). The R-D function evaluation was implemented by making a call to H.264 encoder x264 revision 537, which was modified to accept a list of QPs for the frames. We used 5, 18, 28, and 100 frames from the beginning of 5 different standard video sequences in the experiments, and intra only, IPPP, IPBP, and IPBBP frame type formats. Although not all combinations were tried, especially

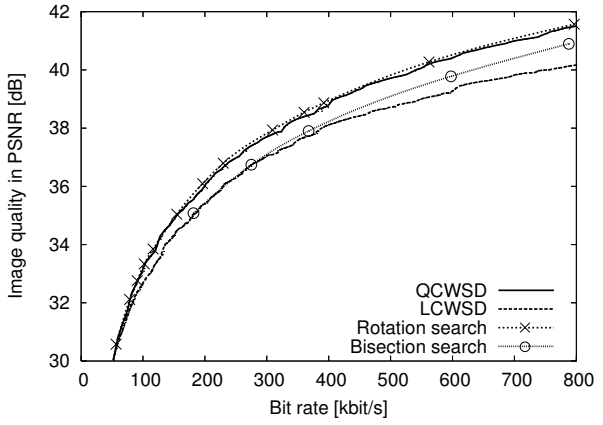
[1] Available from http://tuukkat.thruhere.net/rel/icassp07/



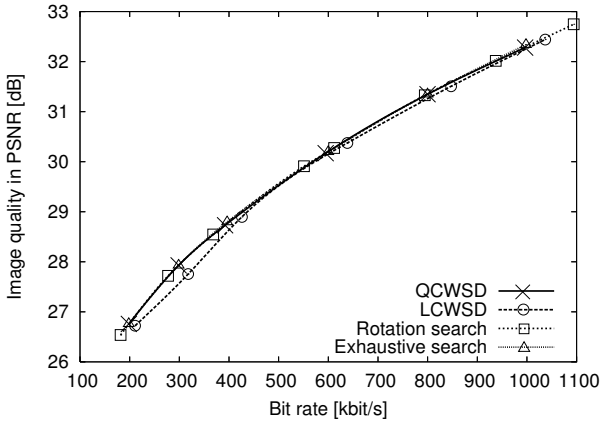
(a) Rate-distortion performance, Tennis, IPBBP, 100 frames



(b) Optimization time, Tennis, IPBBP, 100 frames



(c) Rate-distortion performance, Foreman, IPBBP, 100 frames



(d) Rate-distortion performance, Football, IPPP, 5 frames

Fig. 1. Comparison of R-D optimization algorithms.

at 100 frames, the running time for the experiments on AMD Athlon 2 GHz CPU was still several months.

Some typical results are shown in Fig. 1. As the QCWSD and the LCWSD algorithms generate all achievable R-D points in one run, the corresponding curves in the figures are continuous (except for the 5 frame case, in which only a few points were plotted from the results). For other algorithms, the resulting R-D points are plotted in the figures and a line is interpolated between the points.

Both the QCWSD and the RS reach very nearly the same performance suggesting that both achieve near optimum. In the 5 frame case, a comparison to the exhaustive search algorithm confirms this. Since the LCWSD approximates the slope $\partial D_{\text{tot}}/\partial R_{\text{tot}}$ and the bisection search uses the same QP for all frames, they give significantly worse results than the near-optimal algorithms. Nevertheless, the RS is staggeringly efficient compared to either the QCWSD or the LCWSD, needing typically over 99% less computation than the QCWSD at 100 frames and being faster than even the LCWSD, as demonstrated in Fig. 1b. With shorter sequences the difference is smaller, but already at only 5 frames the RS requires 90% less time than the QCWSD. The bisection search is the most efficient, requiring roughly only about ten calls to the encoder, but it falls seriously behind from the RS and the QCWSD in quality, especially when several frame types (that is, both P and B frames) are used.

Although we did not analyze the RS algorithm complexity rigorously, the number of required encoder calls appears

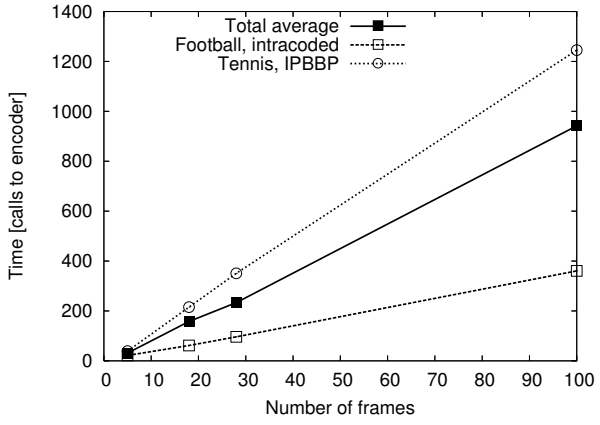


Fig. 2. Complexity of the rotation search.

Sequence name	Search method	Quality (PSNR)	Checking points
Tennis	HEXBS	23.71	16.9
	SDS	23.30	7.8
	RS	22.90	7.0
Foreman	HEXBS	26.48	16.5
	SDS	26.43	7.4
	RS	26.36	6.4
Mobile	HEXBS	22.30	15.2
	SDS	22.29	5.6
	RS	22.28	5.3
Flower	HEXBS	23.64	16.6
	SDS	23.78	6.7
	RS	23.46	6.2

Table 1. Comparison of motion estimation algorithms.

to be quite linear in the number of frames, as shown in Fig. 2, which plots the number of encoder calls at the four different number of frames for two specific cases and as an average over eight different test cases. The initialization stage with the GSS amounts very little to the total time: it requires at most 12 calls to the encoder, independent from the number of frames. The RS itself requires at least $2N + 1$ calls to the encoder to make sure that it has achieved a local minimum. The total number depends on the shape of the multidimensional Lagrangian function, and is difficult to predict theoretically.

ME was implemented using the RS along with the hexagon-based search (HEXBS) [7] and the small diamond search (SDS) [5] algorithms. The motion estimation and compensation was then run for four standard sequences, each containing 100 frames. The results are shown in Table 1. The number of checking points is reduced by 7–13% compared to the SDS with 0.01–0.40 dB degradation in motion compensated image PSNR.

4. CONCLUSIONS

We presented a new fast algorithm for finding local optimum from functions with a high-dimensional and discrete-valued parameter vector. We applied the algorithm for video frame-level R-D optimization by converting rate constrained distortion minimization into an unconstrained Lagrangian function, finding a good approximate initial location with the GSS, and then finding a local optimum using the new search algorithm. The results show that the algorithm finds very nearly the optimum QP vector but is much faster than some other nearly optimal algorithms such as the QCWSD.

While the algorithm suits best for optimizing high-dimensional functions, it is also useful in less dimensions. As a proof, we implemented ME based on the RS algorithm and obtained a good speed-quality tradeoff compared to the other fast ME algorithms.

Although the method is too slow to be used for R-D optimization in many real-time encoders, as it requires multiple coding of the video sequence for gathering the R-D data, R-D models [2] could be used to avoid the encoding step and to speed up the search significantly. In the current form, the algorithm is still useful for offline encoding and benchmarking.

5. REFERENCES

[1] A. Ortega and K. Ramchandran, "Rate-distortion methods for image and video compression," *IEEE Signal Processing Magazine*, vol. 15, no. 6, Nov. 1998, pp. 23–50.

[2] L.-J. Lin and A. Ortega, "Bit-rate control using piecewise approximated rate-distortion characteristics," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 8, no. 4, pp. 446–459, Aug. 1998.

[3] Y. Sermadevi and S. Hemami, "Efficient bit allocation for dependent video coding," in *Proc. Data Compression Conference*, Mar. 2004, pp. 232–241.

[4] Y. Sermadevi, J. Chen, S. Hemami, T. Berger, "When is bit allocation for predictive video coding easy?" in *Proc. Data Compression Conference*, Mar. 2005, pp. 289–298.

[5] W. Choi, J. Lee, S. Yang, and B. Jeon, "Fast motion estimation and mode decision with variable motion block sizes", *Proc. SPIE Visual Comm. Image Proc.*, vol. 5150, pp. 1561–1572, Jun. 2003.

[6] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, 1992.

[7] C. Zhu, X. Lin, L.-P. Chau, "Hexagon-based search pattern for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12. no. 5, Apr. 2002.