



Lefol, D., & Bull, D. R. (2006). Mode refinement algorithm for H.264 inter frame requantization. In 2006 IEEE International Conference on Image Processing, Atlanta, GA, United States. (pp. 845 - 848). Institute of Electrical and Electronics Engineers (IEEE). 10.1109/ICIP.2006.312534

Link to published version (if available):
[10.1109/ICIP.2006.312534](https://doi.org/10.1109/ICIP.2006.312534)

[Link to publication record in Explore Bristol Research](#)
PDF-document

University of Bristol - Explore Bristol Research

General rights

This document is made available in accordance with publisher policies. Please cite only the published version using the reference above. Full terms of use are available:
<http://www.bristol.ac.uk/pure/about/ebr-terms.html>

Take down policy

Explore Bristol Research is a digital archive and the intention is that deposited content should not be removed. However, if you believe that this version of the work breaches copyright law please contact open-access@bristol.ac.uk and include the following information in your message:

- Your contact details
- Bibliographic details for the item, including a URL
- An outline of the nature of the complaint

On receipt of your message the Open Access Team will immediately investigate your claim, make an initial judgement of the validity of the claim and, where appropriate, withdraw the item in question from public view.

MODE REFINEMENT ALGORITHM FOR H.264 INTER FRAME REQUANTIZATION

Damien Lefol and Dave Bull

University of Bristol¹

ABSTRACT

The latest video coding standard H.264 has been recently approved and has already been adopted for numerous applications including HD-DVD and satellite broadcast. To allow interconnectivity between different applications using H.264, transcoding will be a key factor. When requantizing a bitstream the incoming coding decisions are usually kept unchanged to reduce the complexity, but it can have a major impact on the coding efficiency. This paper proposes a novel algorithm for mode refinement of inter prediction in the case of requantization of H.264 bitstreams. The proposed approach gives a comparable quality to a full search for a fraction of its complexity by exploiting the statistical properties of the mode distribution and motion vector refinement.

Index Terms— Video signal processing, video codecs

1. INTRODUCTION

The new standard H.264 [1] is already successful thanks to the variety of scenarios that it can cover and the high quality of video it can deliver even at low bitrates. Recently it has been provisionally approved as one of the standards for HD-DVD and many broadcasters plan to use it to deliver satellite video.

A large amount of research is ongoing in H.264 which is thought to replace MPEG-2 in the coming years. The applications using H.264 will range from multimedia content delivery on mobile handset to High Definition television broadcasting. To allow such diversity in the video broadcasting, it will be necessary to have means of adapting the video to the distribution channel. One solution would be to store only the highest quality bitstream on the server side and to transcode the bitstream depending on the customers needs.

Many algorithms have been developed for the requantization of video in the last decade. Some of these algorithms, such as the Cascaded Pixel Domain Transcoder (CPDT) [2] and the Fast Pixel Domain Transcoder (FPDT) [3], [4], have been used successfully in many practical

applications [5], [6]. It is possible to adapt these algorithms with some changes to the new H.264 standard.

As demonstrated in previous work [7], the FPDT algorithm cannot be used with H.264 bitstream, and the CPDT quality can be significantly lower than a full decode and reencode. This difference is mainly due to the large number of new tools introduced by H.264. The compression efficiency of this new standard is maximal only when all modes are used. When requantizing H.264 bitstream with CPDT, the encoding decisions of the incoming bitstream are kept to reduce the complexity. This implies that the transcoded video uses sub-optimal encoding parameters. A mode refinement algorithm is needed to improve the coding efficiency of the transcoder while reusing as much of the incoming information as possible. The mode refinement algorithm presented in this paper allows the choice of a better prediction mode and motion vector (MV) without having to do a full search in the case of inter frame requantization. The simulation results show that it is possible to have a quality close to a full search while saving more than 85% of the complexity.

Section 2 of this paper will give a quick overview of the requantization algorithm used. The limitation of this algorithm and the proposed mode refinement algorithm are also described together with different possible tuning to balance quality and complexity. Simulation results are given in Section 4. Finally, section 5 concludes the paper.

2. MODE REFINEMENT ALGORITHM

2.1. Requantization algorithm

The requantization algorithm used in this paper is a Cascaded Pixel Domain Transcoder (CPDT) adapted to H.264 as described in [7]. The transcoding complexity is kept low by reusing the encoding information from the incoming bitstream. This approach yields good results when the transcoded bitrate is close to the original bitrate, but for large differences the quality drops. This is due to a sub-optimal use of H.264 encoding macroblock (MB) modes.

The increase of bitrate due to this loss of compression efficiency can be as high as 300% when a very high bitrate needs to be transcoded to lower bitrates (Cf. figure 5). To

¹ This work is funded by 3CRL ROAM4G

compensate for this loss it is necessary to change at least some of the incoming encoding decision. The algorithm presented below achieves an improvement in the compression efficiency of the transcoder up to a level close to a full decode and recode strategy for a fraction of its complexity.

2.2. Mode refinement strategies

One of the problems of mode refinement is to decide whether to refine a macroblock or not since this could lead to poor quality or higher complexity. A first possible complexity simplification is to limit the refinement to MB partition larger than 8x8. Joch shows in [8] that using a MB size of 8x8 and above has only a small impact on the compression efficiency of H.264. A first refinement strategy, *refine*, is to try every macroblock mode larger than 8x8, including intra 16x16. The refined modes are then compared to the incoming mode and the best one is kept.

The statistics obtained from bitstreams requantized at different bitrates show that small size partitions (8x8 and under) tend to become larger when the bitrate decreases. Figure 1 shows the statistical distribution of modes for P frames obtained when encoding a video at three different quantization parameters. It can be noticed that the distribution of mode is highly dependant on the quantization parameter used. Small MB partition sizes, such as intra 4x4 and P 8x8 or smaller, tend to become larger when the QP increases. Note that the bar representing P 16x16 also includes skip MB.

Starting from that property, new mode refinement strategies based on statistical properties can be developed. A similar approach was used in [9] to design a complexity scalable transcoder for H.263. In our case, two limited sets of refinement, *refine set 1* and *refine set 2*, have been defined using the mode distribution statistics. Table 1 gives the type of mode refinement tried depending on the incoming MB type. Within the table a "1" or a "2" means that the corresponding mode is tried for refinement for this incoming MB type in set 1 or set 2 respectively.

The transcoder can also extract key information from the incoming bitstream. For instance it is possible to reuse the reference frame decision or the motion vector information.

The refinement strategies described in this paper decide which frame to use as a reference by taking the most probable one from the incoming bitstream. For instance if an incoming 8x8 MB is refined into a 16x16 one, the four reference frames values from the incoming MB are used to decide only one reference frame for the refined mode.

Refine \ Input	Intra 16x16	P 8x16 and 16x8	P 16x16	Skip
Intra 16x16			1	1
Intra 4x4	1,2		1	1,2
P 8x8 or smaller		1	1,2	1,2
P 8x16 and 16x8			1,2	1,2
P 16x16				1,2

Tab 1. Definition of the limited set for mode refinement

The same approach can be used for MV. Using the previous example the four motion vectors from the incoming bitstream can be merged together to provide one MV for the refined mode. This technique is commonly used in spatial resolution reduction where multiple incoming MVs need to become only one. Previous work shows that using a median of the incoming motion vectors usually yields the best results [10]. The resulting motion vector can be refined using a small size refinement window [11]. This method defines a third mode refinement strategy, $x + \frac{1}{4} \text{ refine}$, where x is the size of the full pel refinement window used and $\frac{1}{4}$ means that a quarter pel refinement is done around the full pel position. Note that when x is null the algorithm is simply called $\frac{1}{4} \text{ refine}$.

4. RESULTS

Simulations were undertaken to assess the complexity and quality of each of these mode refinement strategies. The bitstream to transcode is composed of three concatenated CIF sequences. The first 60 frames are from "Pedestrian", frames 60 to 120 are from "Tractor" and the last 60 frames are from "Toys". These sequences were selected to represent a wide range of possible scenarios and concatenated to simulate a normal consumer environment where scene changes will occur regularly.

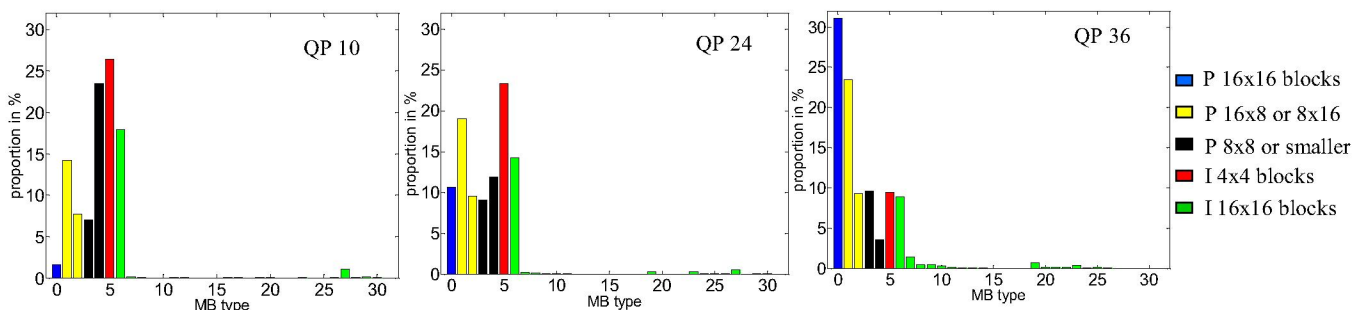


Fig 1. Statistical distribution of mode for a P frame at different QP (mode numbered as defined in [1])

The first sequence contains multiple occlusions and rapid movement, the second a tracking camera and high texture and the third, complex motions and uniform areas. The bitstream has been encoded at 30 frames per second with one intra frame every 30 frames and a group of pictures containing only P inter frame. The original sequence is encoded in H.264 using the reference software JM8.5 at a QP of 10. This sequence is then transcoded at different bitrates using our transcoder. The intra frames contained in the bitstream are transcoded using the threshold refinement algorithm presented in [12]. In the case of full decode and recode, even for the limited set, intra frames are encoded using intra 4x4 and 16x16 with a full search.

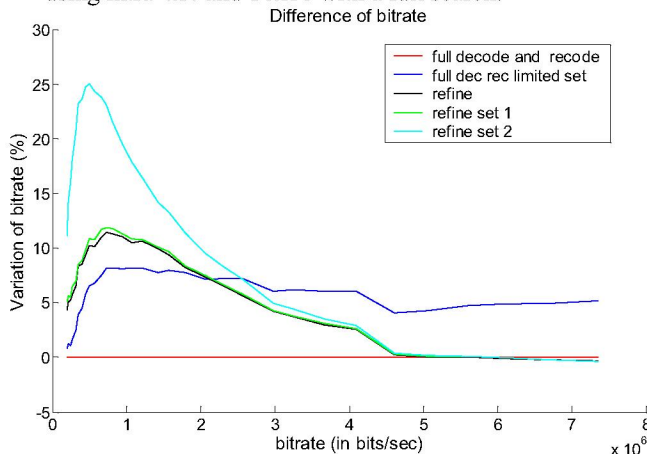


Fig 2. Compression efficiency of strategy 1 and 2

Figure 2 shows the variation of bitrate compared to a full decode and recode for *refine* and *refine set 1* and *set 2*. A full decode and recode with a limited set of MB modes is also plotted. This curve represents the case of the input bitstream being completely decoded and re-encoded with JM8.5 allowing only partition sizes larger than 8x8 for P frames (i.e. P16x16, P16x8, P8x16 and I16x16). It can be observed that the performances of *refine* and *refine set 1* are similar whereas using *refine set 2* has a significant cost on the compression efficiency. The *full decode and recode limited set* approach gives a worse result than *refine* for high bitrate stream as it cannot use a small MB partition size whereas *refine* can keep the incoming mode. On the other hand for low bitrate the full recode performs slightly better as it uses a more complete search by testing all reference frames and testing intra modes for I frames. Table 2 gives a complexity comparison of these algorithms. For a fair comparison the transcoder and *full decode recode* use the same type of ME, thus encoder optimization such as fast motion estimation can be implemented in the transcoder, keeping the complexity gap intact.

Figure 3 shows the comparison of compression efficiency for $x + 1/4$ *refine* using different refinement window sizes x . As expected the compression efficiency increases with the size of the refinement window. This goes

together with a significant increase in complexity. A good balance between quality and complexity can be to use a window size around 4 pixels. Any window smaller than 3 gives mediocre quality results while windows larger than 6 do not improve the compression efficiency significantly but have a significant impact on complexity. When comparing the results from the three strategies the refinement window of size 4 seems a good candidate as it gives results close to *refine* and *refine set 1* for only a fraction of their complexity. It is possible to combine the gains obtained in complexity reduction by the statistical study to the one obtained with the restricted window refinement.

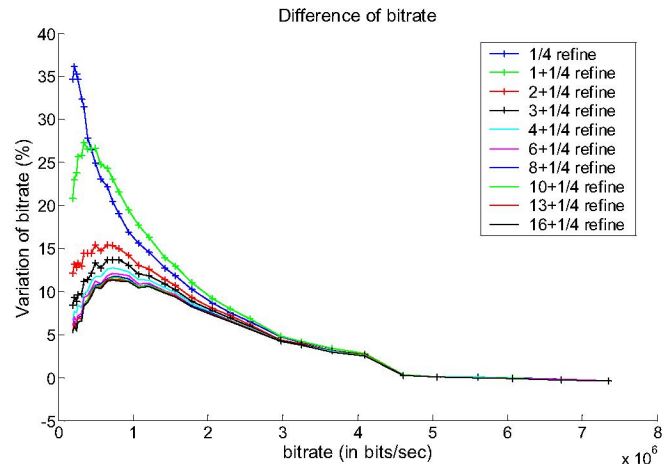


Fig 3. Compression efficiency of strategy 3

Figure 4 compares the compression efficiency of refinement using windows of size 3 to 6 combined with a limited refinement set of type 1, $x + 1/4 + \text{set refine1}$. It can be seen that using a refinement window of size 6 combined with set 1 gives better results than using a refinement window of size 4 without limited set. Moreover the complexity of $6 + 1/4 + \text{set1 refine}$ is lower than $4 + 1/4 \text{ refine}$. Combining strategy 2 and 3 is clearly better.

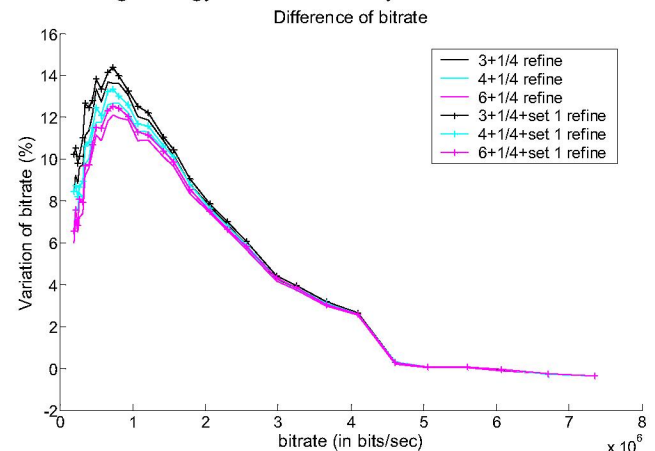


Fig 4. Compression efficiency of strategy 3 + set 1

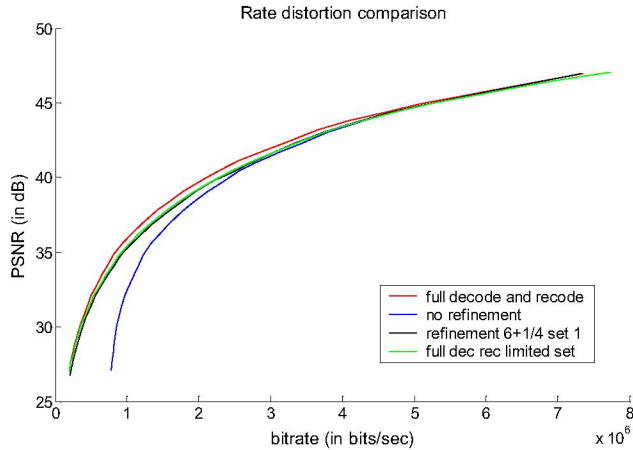


Fig 5. Rate distortion comparison

The plot presented in figure 5 compares the rate distortion of four techniques. The full decode and recode full search, *full decode and recode*, and limited set, *full dec rec*, the refinement using a window of size 6 combined with set 1, *6+1/4+set 1 refine*, and the no refinement plot, *no refine*. The displayed curves show that the difference between no refinement and the three other techniques is important, but the difference between the refined rate distortion and any of the two full decode and recode is small. On the other hand table 2 shows that the average complexity of the refinement strategy used *6+1/4+set 1 refine*, is only 23.4% of a full refine strategy, *refine*. Thus it is possible to have a quality close to a full decode and recode for only a fraction of its complexity when using our algorithm.

Type of refinement	Average time in ms per frame	Complexity (%)
Refinement	1249.4	100
No refinement	63.649	0
Refinement + set 1	943.71	74.2
Refinement + set 2	823.98	64.1
1/4 refine	315.13	21.2
1 + 1/4 refine	317.31	21.4
2 + 1/4 refine	342.08	23.5
3 + 1/4 refine	352.49	24.4
4 + 1/4 refine	382.86	26.9
6 + 1/4 refine	468.76	34.2
8 + 1/4 refine	577.41	43.3
10 + 1/4 refine	720.93	55.4
13 + 1/4 refine	991.35	78.2
16 + 1/4 refine	1326.3	106.5
3 + 1/4 + set 1 refine	257.44	16.3
4 + 1/4 + set 1 refine	272.35	17.6
6 + 1/4 + set 1 refine	341.15	23.4
dec rec limited set	N/A	≈200

Tab 2. Complexity of the different mode refinement strategies

Note that the complexity of the *full decode and recode limited set* is dependant on the number of reference frames

used. In our case it uses two reference frames giving a complexity roughly twice the one of *refine*. The value given in table 2 for *dec rec limited set* is just an indication as the cascaded decode and recode is done with the reference software and not with our transcoder, thus making it hard to compare running time measurements.

5. CONCLUSION

H.264 provides an efficient compression standard for video coding, but when requantizing H.264 bitstream its efficiency can be seriously limited if using non optimal macroblock mode and motion vectors. When the incoming coding information is reused without mode refinement, the bitrate increase can be as high as 300% compared to a full recode. The mode refinement algorithm presented here provides an efficient tool to keep the quality and bitrate of the transcoded bitstream at its maximum while reducing the complexity by more than 85% compared to a full search approach.

REFERENCES

- [1] ISO, "ISO/IEC 14496-10 AVC | ITU-T Rec. H.264 International Standard for Advanced Video Coding," 2003.
- [2] A. Vetro, C. Christopoulos, and H. Sun, "Video transcoding architectures and techniques: An overview," IEEE Signal Processing Magazine, vol. 20, pp. 18-29, 2003.
- [3] P. A. A. Assuncao and M. Ghanbari, "Frequency-domain video transcoder for dynamic bit-rate reduction of MPEG-2 bit streams," IEEE Transactions on Circuits and Systems for Video Technology, vol. 8, pp. 953-967, 1998.
- [4] J. Youn, M.-T. Sun, and J. Xin, "Video transcoder architectures for bit rate scaling of H.263 bit streams," presented at Proceedings of the 1999 7th International Multimedia Conference - ACM MULTIMEDIA '99, Oct 30-Nov 5 1999, Orlando, FL, USA, 1999.
- [5] P. N. Tudor and O. H. Werner, "Real-time transcoding of MPEG-2 video bit streams," presented at Proceedings of the 1997 International Broadcasting Convention, Sep 12-16 1997, Amsterdam, Neth, 1997.
- [6] ACTS, "Advanced television at low bit rates and networked transmission over integrated communication systems," Online: web-site: <http://www.bbc.co.uk/atlantic/>, 1996.
- [7] D. Lefol, D. Bull, N. Canagarajah, "Performance Evaluation of Transcoding Algorithms for H.264", IEEE Transactions on Consumer Electronics, 52 (1) (2006) 66-73.
- [8] A. Joch, F. Kossentini, and P. Nasiopoulos, "A performance analysis of the ITU-T draft H.26L video coding standard," 12th International Packetvideo Workshop, April 2002.
- [9] J. Yeh and G. Cheung, "Complexity Scalable Mode-Based H.263 Video Transcoding", presented at IEEE International Conference on Image Processing, V 1, p 169-172, 2003
- [10] Y. P. Tan and H. Sun, "Fast Motion Re-Estimation for Arbitrary Downsizing Video Transcoding using H.264/AVC Standard", IEEE Transaction on Consumer Electronics, V 50, No 3, p 887-894, August 2004.
- [11] J. Youn, M.-T. Sun, and C.-W. Lin, "Motion Vector Refinement For High Performance Transcoding," IEEE Transactions on Multimedia, vol. 1, No. 1, March 1999.
- [12] D. Lefol, D. Bull and N. Canagarajah, "Mode Refinement Algorithm for H.264 Intra Frame Requantization", accepted at iscas 2006, Kos, Greece.