

Improved Refinement Search for H.263 to H.264/AVC Transcoding Based on the Minimum Cost Tendency Search

Chi-Wang Ho Oscar C. Au S.-H. Gary Chan Hoi-Ming Wong Shu-Kei Yip

The Hong Kong University of Science and Technology,
Clear Water Bay, Kowloon, Hong Kong, China

Email: {jodyho, eeau, gchan, hoimingw, sukkiyip}@ust.hk

Abstract—An improved refinement search method for transcoding from H.263 to H.264/AVC is proposed in this paper. Many existing motion re-estimation methods refine the input motion vector (MV) with a small search range, which is usually input MV biased. Motion estimation (ME) in H.263 usually does not consider the rate required for coding the MV, and hence, the input MV may incur a large cost in H.264/AVC. To overcome this problem, we introduce a refinement search method, called Minimum Cost Tendency Search (MCTS), which takes the difference between the cost functions for ME in H.263 and H.264/AVC into consideration. The input MV and the predictor MV are used as two anchor points. The proposed MCTS starts searching from the anchor point with a higher cost to another. Finally, the best point is chosen as the center for further refinement. The performance of MCTS is evaluated by comparing with full search, FME in JM software and refinement scheme using small diamond pattern around the input MV (RSD). Experimental results show the proposed MCTS performs more stable than FME and RSD over a wide range of output video quality.

I. INTRODUCTION

H.264/AVC is the newest international video coding standard [1]. With its advanced coding tools, it outperforms all of the existing standards, such as H.263+ [2] and MPEG-4 [3], in terms of both quality and coding efficiency. Therefore, H.264/AVC is a strong candidate for a wide range of applications in the future.

Video transcoding is the conversion of one encoded video into another encoded video [4], [5], which may have different format, bit-rate, resolution, etc. This allows the pre-coded videos to convert from the existing standards to H.264/AVC, and takes the potential advantages of H.264/AVC. However, H.264/AVC transcoding raises a number of new issues [6], so it cannot perform efficiently in the transform domain. Therefore, the most straightforward way to perform H.264/AVC transcoding is to decode the input video fully and re-encode the reconstructed raw frames. To reduce the computational complexity, motion vector (MV) refinement with a small window size around the input MV could be used instead of the exhaustive search. Some well-known refinement search patterns, such as diamond search pattern and hexagonal search pattern, are commonly used to reduce the complexity. There are many research works on fast motion re-estimation during transcoding from different standards to H.264/AVC [7]–[9]. They use the input MV as the predictor or the center of the refinement search.

In this paper, a new refinement search method, called Minimum Cost Tendency Search (MCTS), for transcoding the video from H.263 to H.264/AVC is proposed. Based on the difference between the cost functions for motion estimation (ME) in H.263 and H.264/AVC, the optimal MV for H.264/AVC is probably located in a region bounded by the input MV and the predictor MV. The proposed MCTS picks some points between the input MV and the predictor MV, and calculates their cost. Then, a further refinement is performed using

This work has been supported in part by the Innovation and Technology Commission (projects no. ITS/122/03 and GHP/033/05) and the Research Grant Council (DAG04/05.EG34) of the Hong Kong Special Administrative Region, China.

the small diamond pattern centered at the point with minimum cost of the set of the examined points.

The rest of this paper is organized as follows. We present in Section II the overview of the cost functions for the ME in H.263 and H.264/AVC followed by the proposed MCTS. Experimental results illustrating the performance of the proposed method is presented in Section III and end with concluding remarks.

II. PROPOSED MINIMUM COST TENDENCY SEARCH (MCTS)

In H.263 and H.264/AVC, they use different cost functions for ME. The cost function of H.263 aims to minimize the distortion only, whereas that of H.264/AVC attempts to consider both the distortion and rate for coding the MVs. Therefore, simple refinement using the input MV probably is not good enough when the cost of the input MV is considerably high in H.264/AVC. We will first give a brief overview of the cost functions for ME in H.263 and H.264/AVC. Then, the proposed MCTS will be presented and discussed in details.

A. Cost function for Motion Estimation

We first denote (x, y) as the coordinates of the current macroblock (MB). With a given search range, S is a set of candidate MVs within the search range and mv^i is the candidate MV i where $mv^i \in S$. The distortion, $D(\cdot)$, can be defined as a function that takes the reference frame, the current MB at (x, y) and the candidate MV mv^i as the inputs.

1) *H.263*: The ME in H.263 can be formulated as follows:

$$mv^* = \arg \min_{mv^i \in S} D(\cdot) \quad (1)$$

In H.263, $D(\cdot)$ is simply the sum of absolute difference (SAD). Therefore, the optimal MV, mv^* , is the MV which gives the minimum SAD among all candidate MVs in S .

2) *H.264/AVC*: However, H.264/AVC supports motion compensation with variable block sizes. A large number of combinations within each MB is possible and a separate MV is required for each partition. With a smaller block size, we expect to have smaller residual energy but larger number of bits required for coding the MVs and partition(s). The problem of choosing the best MV is formulated as a rate-constrained optimization problem and the optimal MV is the MV which minimizes the following Lagrangian cost function,

$$J(\cdot) = D(\cdot) + \lambda_{motion} R(\cdot) \quad (2)$$

This takes the distortion and the rate required for coding the MVs into consideration. Therefore, the ME in H.264/AVC can be written as,

$$mv^* = \arg \min_{mv^i \in S} J(\cdot) \quad (3)$$

In the above equation, λ_{motion} is a Lagrangian multiplier imposing the rate constraint of the MVs which is QP dependent. $R(\cdot)$ represents the number of bits required for coding the motion information, it depends on both the input and predictor MV. In H.264/AVC, $D(\cdot)$

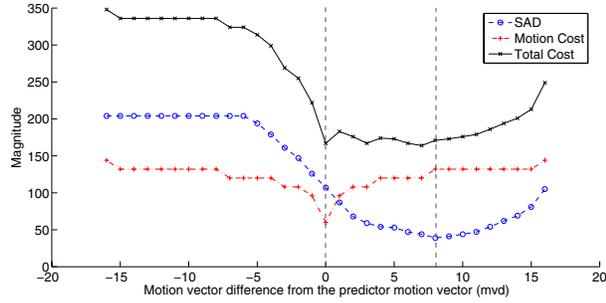


Fig. 1. The motion cost, SAD and total cost against the displacement from the predictor motion vector (mvd) in one-dimensional space.

can be either the SAD or the sum of absolute difference of Hadamard-transformed coefficients (SATD). In the following discussion, we assume that $D(\cdot)$ is the SAD, which is the one used for ME in H.263. The proposed method should also be applicable to the SATD with minor adjustment.

B. Proposed Minimum Cost Tendency Search (MCTS)

According to Eq. (1) and (2), the major difference between the cost functions for ME in H.263 and H.264/AVC is $\lambda_{motion}R(\cdot)$, which imposes the rate constraint of the MVs in H.264/AVC. Its value always increases from the predictor MV, which is the minimum point of $\lambda_{motion}R(\cdot)$, outward.

For the distortion function, $D(\cdot)$, the relationship between the SAD calculated in the H.263 front-encoder, $D_{h263}(\cdot)$, and the SAD calculated in the H.264/AVC transcoder, $D_{h264}(\cdot)$, can be expressed in the following equation,

$$D_{h264}(\cdot) = D_{h263}(\cdot) + e, \quad (4)$$

where e is the error introduced by encoding. $D_{h263}(\cdot)$ is the SAD between the current MB from the original source and the reconstructed frame after H.263 encoding. $D_{h264}(\cdot)$ is the SAD between the current MB from the decoded H.263 video and the reconstructed frame after H.264/AVC encoding. Here we first assume that the magnitude of e is very small and negligible. In general, this is valid when the quality of both the input and output video is high. According to Eq. (1), the $D_{h263}(\cdot)$ corresponding to the input MV is the minimum within a given search range in the H.263 encoder. Under the assumption that e is negligible, in the H.264 transcoder, the $D_{h264}(\cdot)$ corresponding to the input MV is also the minimum within the same search range.

Based on our assumption, the minimum points for $D(\cdot)$ and $\lambda R(\cdot)$ in Eq. (2) are known. If we make another assumption that $D(\cdot)$ is a smooth surface with a single minimum at the input MV, the Lagrangian cost, $J(\cdot)$ in Eq.(2), should be located in the region between the input MV and the predictor MV. This is illustrated by a one-dimensional example shown in Fig. 1.

There are three lines in Fig. 1, from top to bottom, represent the total cost $J(\cdot)$, the distortion $D(\cdot)$, or simply the SAD, and the motion cost $\lambda_{motion}R(\cdot)$. The x-axis represents the MV difference (mvd) between the candidate MV and the predictor MV, and the y-axis represents the magnitude of the corresponding cost. As the curves of both $D(\cdot)$ and $\lambda_{motion}R(\cdot)$ increase from their minimum position to both left and right, the sum of these two values should always increase to the left of the minimum point of $\lambda_{motion}R(\cdot)$ and to the right of the minimum point of $D(\cdot)$ which indicates by the dotted line in the figure. Therefore, the minimum point of $J(\cdot)$ must be inside the region bounded by the minimum point of $D(\cdot)$ and that of

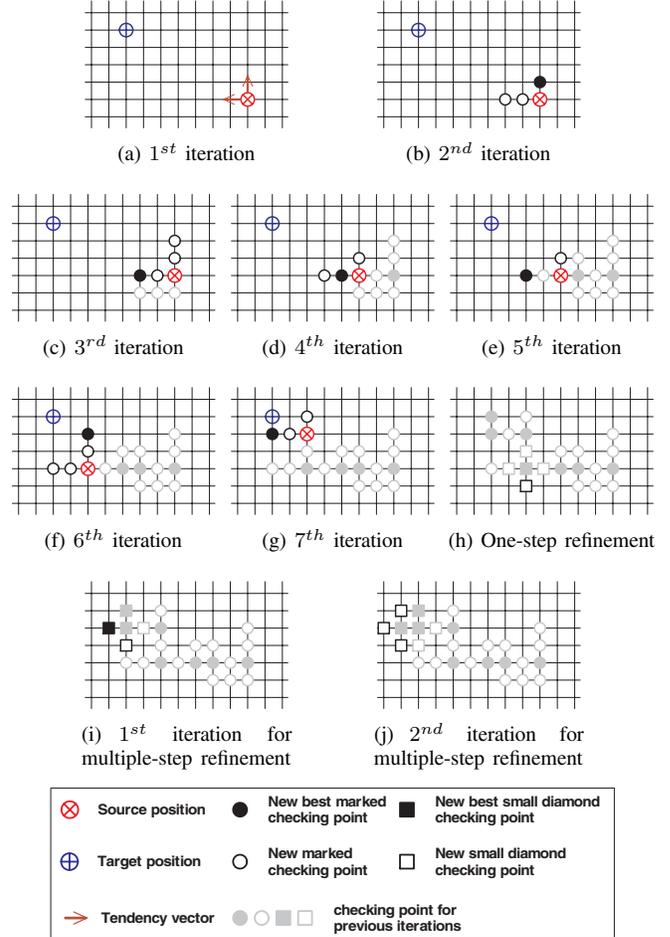


Fig. 2. An example of the proposed MCTS - (a) shows the position of the source position and the target position. Two arrows indicates \hat{v}_{tend}^x and \hat{v}_{tend}^y . (b)-(g) illustrate how to iterate from one point to another based on the rule stated in Step 4. (h) shows the case of the one-step small diamond refinement. (i)-(j) shows the case of multiple-step small diamond refinement.

$\lambda_{motion}R(\cdot)$. For this reason, we propose to use the input MV from H.263 (mv_{h263}) and the predictor MV (pmv) as two anchor points. Then, several iterations are performed between the two anchor points before having the refinement with the small diamond pattern.

However, in the real situation, the cost surface of $D(\cdot)$ usually depends on the video content, it is not a smooth surface with a single minimum and hard to represent accurately by using models. Therefore, there is no easy way to find the optimal MV for Eq. (2) even though the first assumption is valid. Indeed, the magnitude of e is not necessary to be small, it depends on the quality of both the input and output video. In the proposed MCTS, a greedy approach is used. It tries to search toward the direction, which probably has a better point. The proposed MCTS can be briefly summarized as the following steps,

- Step 1.** Check whether mv_{h263} and pmv is equal or very close to each other. If the distance between them is less than one pixel, go to Step 8 to perform refinement using mv_{h263} as the center.
- Step 2.** Calculate the costs corresponding to the input MV and the predictor MV, $J(mv_{h263})$ and $J(pmv)$, using Eq. (2). If $J(mv_{h263})$ is larger than $J(pmv)$, mv_{h263} is set as source

position and pmv is set as target position. Otherwise, pmv is set as source position and mv_{h263} is set as target position.

Step 3. Find the tendency vector, $\vec{v}_{tend.}$, pointing from the source position to the target position, calculate its magnitude in each direction, $|\vec{v}_{tend.}^x|$ and $|\vec{v}_{tend.}^y|$, and their normalized vector $\hat{v}_{tend.}^x$ and $\hat{v}_{tend.}^y$. They are used to indicate the direction of the search.

Step 4. Choose several new search positions, at most two in each direction, around the source position based on the results of the previous iteration and the magnitude of the tendency vector as follows:

- If $|\vec{v}_{tend.}^x| > |\vec{v}_{tend.}^y|$, mark two positions in $\hat{v}_{tend.}^x$ direction and one position in $\hat{v}_{tend.}^y$ direction. Otherwise, mark one position in $\hat{v}_{tend.}^x$ direction and two positions in $\hat{v}_{tend.}^y$ direction.
- If the previous iteration is moving in $\hat{v}_{tend.}^x$ direction, mark two positions in $\hat{v}_{tend.}^x$ direction. Otherwise, mark two positions in $\hat{v}_{tend.}^y$ direction.

Step 5. Calculate the cost J of these marked positions and store them in a cost map, M_{cost} .

Step 6. Update the source position to the position with minimum cost among these marked positions, and the tendency vector $\vec{v}_{tend.}$. Go to Step 4 until the source position hits the target position.

Step 7. Set the center of the final refinement to the position with the minimum cost in M_{cost} .

Step 8. Perform either one-step or multiple-step small diamond refinement, depending on the position of the selected center, mv_{h263} and pmv . Finally, obtain the resulting MV.

From Step 4 to 6, the proposed MCTS is forced to find a path between mv_{h263} and pmv and examine all the points on the path. The search starts from the MV with a higher cost. The points are selected for checking based on the conditions described in Step 4. The first condition is used to ensure the target position can be reached because more points are checked in the direction with a larger difference to the target. The second condition attempts to check more points in the direction which tends to have a lower cost based on the results of the previous iteration. Therefore, this greedy approach attempts to move in a direction which is closer to the target and probably with a lower cost. Moreover, by forcing to examine all the points on the path, this can reduce the chance to get trapped in a bad local minimum. Then, the refinement can be started at a better position. This approach comes with a cost of additional complexity which needs to check more points depending on the difference between mv_{h263} and pmv . Fortunately, in general, these two MVs should be close to each other in typical video sequences so that the number of the additional checking points is likely to be small.

Either one-step or multiple-step refinement is used for further refinement. If the position with minimum cost is located somewhere far from mv_{h263} and pmv , this situation seems to match our assumptions discussed previously. So, only one-step refinement is used. Otherwise, if the best position is close to either mv_{h263} or pmv , which has a distance within one pixel, this means that our assumptions may be invalid in this case. Therefore, multiple-step refinement is used and terminates when the minimum is occurred at the center of the diamond pattern. An example in Fig. 2 shows how the proposed MCTS works.

III. PERFORMANCE EVALUATION

We have implemented the proposed MCTS on a cascaded transcoder using H.263 and H.264/AVC reference codec [10], [11].

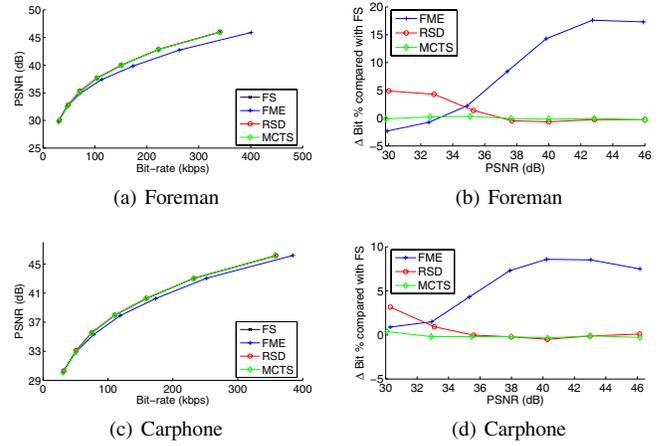


Fig. 3. Comparison of bit-rate and PSNR for different sequences with input QP=28 and transcoded to different QPs. (a) and (c) are the R-D curve. (b) and (d) are the Δ Bit % against PSNR which is used to show the bit fluctuation with the PSNR.

The four test sequences of QCIF (176x144) were precoded using a fixed quantization parameter (QP) and full search. The QP of the first I frame was set to 13 and the QP of the remaining P frames were set to either 18 or 28. Then, each test sequence was transcoded to H.264/AVC with one reference frame, CAVLC, RDO disabled and only P16x16 mode. Since the intra coded MBs in the input video did not contain any motion information, they were either encoded in I4x4 or I16x16 for fair comparison.

Table I shows the results of different algorithms at different QPs. The performance of full search, fast motion estimation in H.264/AVC reference encoder (FME), simple refinement with small diamond pattern around the input MV (RSD) and the proposed MCTS were compared in terms of PSNR, bit-rate and speed.

Based on our simulations, all the fast algorithms show similar average PSNR comparing with full search. In terms of bit-rate, the proposed MCTS performs more stable than FME and RSD over a range of output QPs. In Table I, the percentage increase in bit-rate compared with full search, Δ Bit %, is shown. We can see that, in general, FME has a large increase, 8.40%, in bit-rate when the output QP is small. In contrast to FME, RSD also has 4.30% increase in bit-rate when the output QP is large. They perform differently at different QPs. However, the proposed MCTS has 0.84% increase in bit-rate at different output QPs. Fig. 3 shows a clearer comparison between different algorithms by the R-D curve and the percentage increase in bit-rate compared with full search against the PSNR.

The large increase in bit-rate of FME is probably because of the prior H.263 encoding. According to Eq. (2), the optimal MV for H.264/AVC is based on the distortion and the rate for coding the motion information. When the QP is small, the number of bits for coding the coefficients is dominant, in other words, it would be better to find the position with the minimum distortion. During the H.263 encoding, the residue of the MB was transformed and quantized. The residual energy is usually smaller after quantization, hence, the difference between the current and the reference MB is now also smaller. If the same reference MB can be used in H.264/AVC, this would probably be the reference MB with the minimum distortion. However, in our experiments, full search is used to determine the MV in the H.263 encoder, so the MV field may not be smooth. Since FME does not search every single point within the search range, the point

TABLE I
COMPARISON BETWEEN FULL SEARCH, FME, RSD AND MCTS FOR QCIF SEQUENCES
TRANSCODED FROM H.263 TO H.264 WITH DIFFERENT QPS.

Seq.	QP in	QP out	Full Search		FME			RSD			MCTS		
			PSNR	Bit	Δ PSNR	Δ Bit %	Speed up	Δ PSNR	Δ Bit %	Speed up	Δ PSNR	Δ Bit %	Speed up
akiyo	18	28	38.94	218296	+0.05	+0.30	113.64	-0.02	+0.09	216.39	+0.00	+0.00	215.93
		36	35.02	70072	+0.02	-1.26	154.34	+0.01	+0.59	215.10	-0.01	+0.37	215.55
	28	28	39.38	142888	-0.08	+2.83	128.00	-0.03	+0.07	216.46	+0.00	+0.00	216.39
		36	35.98	60616	+0.15	-0.40	163.16	+0.08	-0.30	216.51	+0.05	+0.07	216.41
foreman	18	28	37.11	1397160	-0.22	+5.87	28.40	+0.00	-0.10	187.46	-0.03	+0.01	157.50
		36	31.71	500312	-0.11	-1.25	35.97	+0.06	+3.06	176.68	-0.05	+0.01	156.08
	28	28	37.66	1039592	-0.25	+8.40	31.46	+0.03	-0.47	188.64	-0.02	-0.05	154.80
		36	32.68	464472	-0.17	-0.74	39.12	+0.15	+4.30	178.78	-0.10	+0.21	154.28
stefan	18	28	34.62	4601192	-0.03	+0.52	22.64	+0.01	+0.31	185.34	-0.02	+0.16	160.81
		36	28.20	1405432	-0.01	-0.08	25.56	+0.02	+2.22	176.48	+0.00	+0.84	162.58
	28	28	35.13	3535888	-0.04	+0.88	23.45	+0.01	+0.01	186.41	-0.02	+0.02	158.31
		36	29.05	1297600	-0.01	-0.20	25.94	+0.02	+1.08	177.99	+0.00	+0.60	160.06
Carphone	18	28	37.54	1550752	-0.08	+3.21	35.75	+0.00	-0.18	194.60	-0.04	-0.19	174.09
		36	31.97	556992	+0.05	+0.33	46.23	+0.08	+2.00	186.30	-0.02	+0.64	174.68
	28	28	37.97	1104520	-0.07	+7.30	40.01	+0.02	-0.22	197.25	+0.02	-0.20	172.51
		36	32.97	500664	-0.04	+1.49	48.85	+0.13	+0.92	189.79	-0.07	-0.19	172.90
Average					-0.05	+1.70	60.16	+0.04	+0.84	193.14	-0.02	+0.14	176.43

corresponding to the input MV may not be checked and selected as the output MV. It cannot take the potential advantage of the prior encoding. When the output QP increases, this effect is diminished because the rate of coding the motion information becomes more significant and the distortion introduced to the reference frame by H.264/AVC encoding also increases.

On the other hand, RSD, which refines the input MV with a simple diamond search, also shows an increasing bit-rate difference comparing with full search when the QP increases. As we mentioned previously, the refinement scheme is usually input MV biased. The MV refined by RSD may suffer from a significant motion vector cost at large QP. This probably affects its performance.

The speed up factor is defined as the ratio of the required number of checking points for the fast algorithm compared to that of full search. With the input MV, the proposed MCTS and RSD are significantly faster than the FME. The difference between the speed of the proposed MCTS and RSD varies depending on the content of the video sequences. The speed is about the same for the sequence with slow motion, such as akiyo. However, for the sequence with median or high motion, such as foreman and stefan, the proposed MCTS takes a few more search points than RSD, on average one more search point is needed for each MB, because the input MV and the predictor MV are not close to each other.

IV. CONCLUSIONS

We have presented the proposed Minimum Cost Tendency Search (MCTS) to refine the input MV extracted from H.263 video which explicitly considers the difference between the cost functions for ME in H.263 and H.264/AVC. We have shown that in H.263-to-H.264/AVC transcoding using the input MVs with refinement may introduce performance degradation when the QP is large. With the proposed MCTS, it significantly avoids the performance degradation with little increases in computational complexity. The experimental results can be used to verify the effectiveness of the proposed MCTS. Specifically, the proposed MCTS can perform more stable compared with FME and simple refinement over a wide range of output video quality, and keep a low computational complexity.

ACKNOWLEDGMENT

This work has been supported in part by the Innovation and Technology Commission (projects no. ITS/122/03 and GHP/033/05) and the Research Grant Council (DAG04/05.EG34) of the Hong Kong Special Administrative Region, China.

REFERENCES

- [1] *Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. H.264/ISO/IEC 14 496-10 AVC)*, Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, May 2003, JVT-G050.
- [2] *Video Coding for Low Bitrate Communication, Recommendation H.263 version 2*, ITU Telecom. Standardization Sector of ITU, Feb. 1998.
- [3] *Coding of Audio-Visual Objects - Part 1: Systems, ISO/IEC 14496-1 International Standard*, MPEG98, Mar. 2000, ISO/IEC JTC1/SC29/WG11 N2501.
- [4] A. Vetro, C. Christopoulos, and H. Sun, "Video transcoding architectures and techniques: an overview," *IEEE Signal Processing Magazine*, vol. 20, no. 2, pp. 18–29, Mar. 2003.
- [5] J. Xin, C.-W. Lin, and M.-T. Sun, "Digital video transcoding," in *Proceedings of the IEEE*, vol. 93, no. 1, Jan. 2005, pp. 84–97.
- [6] I. Ahmad, X. Wei, Y. Sun, and Y.-Q. Zhang, "Video transcoding: an overview of various techniques and research issues," *IEEE Transactions on Multimedia*.
- [7] K.-T. Fung and W.-C. Siu, "Low complexity H.263 to H.264 video transcoding using motion vector decomposition," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, no. 5, May 2005, pp. 908–911.
- [8] G. Chen, Y. dong Zhang, S. xun Lin, and F. Dai, "Efficient block size selection for MPEG-2 to H.264 transcoding," in *Proceedings of the 12th annual ACM International Conference on Multimedia*, no. 1, Oct. 2004, pp. 300–303.
- [9] S.-E. Kim, J.-K. Han, and J.-G. Kim, "Efficient motion estimation algorithm for MPEG-4 to H.264 transcoder," in *Proceedings of the IEEE International Conference on Image Processing*, no. 3, Sept. 2005, pp. 656–659.
- [10] Image Processing Lab, University of British Columbia, "TMN (H.263+) encoder/decoder, version 3.2," Sept. 1997. [Online]. Available: <http://www.ee.ubc.ca/image>
- [11] "H.264/AVC Reference Software JM8.6," Oct. 2004. [Online]. Available: <http://iphome.hhi.de/suehring/tml/>