An Efficient Coding Method for Intra Prediction Mode Information

Kai Zhang[1, 2], *Xiangyang Ji*[1, 2], *Qian Huang*[1, 2], *Debin Zhao*[1], *Wen Gao*[2]

[1]Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100080, China
[2]Graduate University of the Chinese Academy of Sciences, Beijing 100039, China
{kzhang, xyji, qhuang, dbzhao, wgao }@jdl.ac.cn

ABSTRACT

In H.264/AVC, the bits used for the Intra_4x4 prediction mode information usually occupy a high percentage in intra coding. Towards this issue, we present an efficient coding method for the Intra_4x4 prediction mode information. Firstly, a 3-order Markov random field is introduced to model the correlation among neighboring 4x4 blocks at picture level. Secondly, based on the conditional probabilities learned in this model, we build up a context adaptive coding scheme to code the Intra_4x4 prediction mode information. Although the probabilities and the coding scheme are initialized off-line, they can be revised by automatic adjustments. Thus the proposed algorithm is robust to a variety of video sequences. Experimental results demonstrate that the proposed method can obtain a gain up to 0.3 dB in all I-frames coding without involving any serious computational burden.

Index Terms**—**Image/video coding, H.264/AVC, Intra prediction

1. INTRODUCTION

H.264/AVC [1] is a novel international video coding standard which improves coding performance significantly. The new intra coding technique is one of the major innovations in H.264/AVC. It contributes a lot in the coding efficiency, especially in all I-frames coding, which plays an important role in some professional applications. H.264/AVC introduces directional intra prediction in the spatial domain [1]. Several predefined intra-prediction modes are available in the Lagrangian Rate-Distortion Optimization (RDO) approach.

To further improve the performance of intra coding in H.264/AVC, several techniques have been proposed. Fan *et al.* [2] developed novel scanning methods for different directions. Zhang *et al.* [3] introduced multiple intra prediction modes. Yang *et al.* [4] utilized template matching both on the encoder side and on the decoder side. Recently, Shiodera *et al.* [5] proposed a bidirectional intra prediction approach, which changed the scanning order of intra-blocks.

Previous works focused on coding the texture information more efficiently, without considering the mode information. Nevertheless, the bits used for the Intra_4x4 prediction mode information occupy a high percentage of the total bits in intra coding, since the coding method for the Intra_4x4 prediction mode information in H.264/AVC is not efficient enough.

To tackle this problem, we present an efficient coding method for the Intra_4x4 prediction mode information in H.264/AVC. Intuitively, since neighboring blocks hold a strong correlation, we can predict the Intra_4x4 prediction mode of the current block based on the modes of coded neighboring blocks. The proposed method takes advantage of this correlation by adopting several techniques. Initially, the spatial distribution of RD optimal [6] Intra_4x4 prediction modes in a picture is modeled as a Markov random field, where the conditional probabilities are learned off-line. By using these conditional probabilities, a context adaptive approach is applied to code the Intra_4x4 prediction mode information. Under the context adaptive framework, we introduce a new macro-block mode, namely Intra_4x4_Implicity mode, in which Intra_4x4 prediction modes are induced implicitly instead of being encoded explicitly. Since the coding scheme is initialized with data learned off-line on a finite training set, the initial state of the coding scheme may be not suitable for some sequences. For the purpose of robustness, the conditional probabilities and the code tables are updated by automatic adjustments on-line to track the specific characters of sequences being encoded. Experiments indicate that the proposed method compresses the Intra_4x4 prediction mode information more efficiently, thus obtains a considerable gain in coding performance.

The remainder of this paper is organized as follows. In Section 2, we review the Intra_4x4 prediction in H.264/AVC briefly. Section 3 describes the proposed coding method. Experimental results are presented and discussed in Section 4. Section 5 concludes the paper.

2. INTRA_4x4 PREDICTION IN H.264/AVC

Intra_4x4 and Intra_16x16 are two types of luma intra-prediction in H.264 baseline profile and main profile [1]. Intra_4x4 has up to nine prediction modes, namely mode 0 - mode 8. Except the DC mode (mode 2), the other eight modes correspond to different prediction directions as illustrated in Fig. 1.

In H.264/AVC, the Intra_4x4 prediction mode information is coded with a rough prediction from neighboring blocks. Suppose mode M_A and mode M_B represent the Intra_4x4 prediction modes of the left and upper neighboring blocks respectively, then M_p = minimum $\{M_A, M_B\}$ is determined to be the most probable mode for the current block. If the mode of the current block just equals to M_p, the syntax element **prev_intra4x4_pred_mode_flag** with one bit is set to be 1. Otherwise, it is set to be 0 and an additional syntax element **rem_intra4x4_pred_mode** with three bits is conveyed.

The coding method of the Intra_4x4 prediction mode information in H.264/AVC is not efficient enough. As depicted in Fig. 2, the bits for the Intra_4x4 prediction mode information take quite a large part of the total bits in all I-frames coding, especially at low bit-rates.

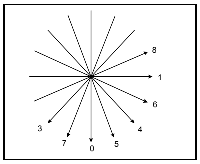
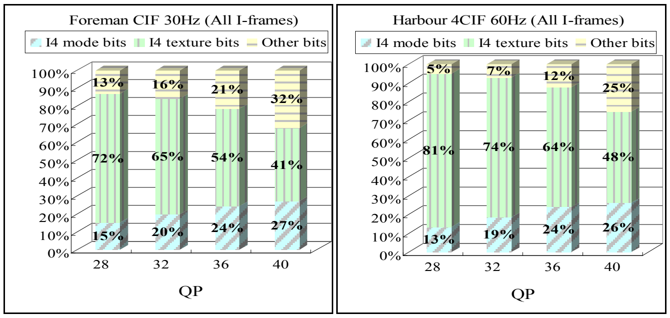


Figure 1. Intra_4x4 prediction modes.

This work was supported in part by National Science Foundation (60833013, 60803068) and Major State Basic Research Development Program of China (973 Program, 2009CB320903)



(A) (B)

Figure 2. The composition of bits in all I-frames coding of sequences 'Foreman' in CIF format (a) and 'Harbour' in 4CIF format (b). 'Other bits' represents the bits used for coding the header information and macroblocks in Intra_16x16 mode.

3. PROPOSED CODING METHOD

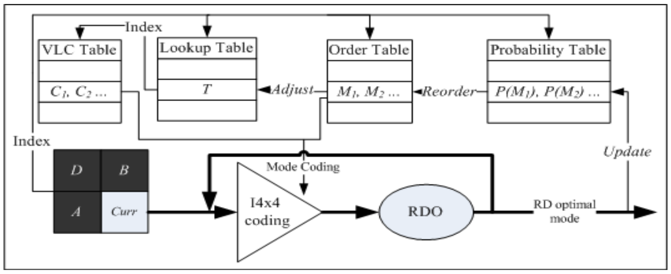


Figure 3. The proposed architecture on the encoder side. '*Curr*' denotes the current 4x4 block. '*A*', '*B*' and '*D*' represent the coded neighboring blocks.

An efficient coding method for the Intra_4x4 prediction mode information is designed in several steps. We mainly describe the method on the encoder side, and it is symmetric on the decoder side. Firstly, based on a Markov random field hypothesis, we do statistics on some video sequences to obtain mode probabilities in different neighboring contexts, and sort the modes in probability descending order. The probabilities and the order of modes are stored in Probability Table and Order Table respectively, which are both indexed by neighboring contexts. Secondly, we utilize a context adaptive approach to code the Intra_4x4 prediction mode information. Instead of coding the mode directly, we choose the position of the mode in Order Table as the symbol word to be coded. Ten types of Huffman code tables with index number 0 – 9 are predefined and stored in VLC Table. In each neighboring context, the index number of the Huffman code table with the shortest expected code length is recorded in Lookup Table. When encoding Intra_4x4 prediction modes in a specific neighboring context, a Huffman code table is selected from VLC Table by the index number stored in Lookup Table. Under the context adaptive framework, we introduce a new macro-block mode, namely Intra_4x4_Implicity mode. In this mode, no Intra_4x4 prediction mode information is coded, and a 4x4 block will hold the most probable mode in its neighboring context. Thirdly, after the RDO procedure chooses the optimal Intra_4x4 prediction mode, an updating process will refine Probability Table, Order Table and Lookup Table to track the characters of sequences being encoded. Fig.3 depicts the architecture of the proposed method on the encoder side.

3.1. Markov Random Field

To characterize the correlation among neighboring blocks, we model the spatial distribution of RD optimal Intra_4x4 prediction modes in a picture as a 3-order Markov random field [7]. In this model, the probability that a mode is the RD optimal one of the current block depends on the RD optimal modes of coded neighboring blocks. Formally, we define

$$\begin{aligned} &P(M_{Curr} \mid (M_A, M_B, M_D)) \equiv \\ &P(Mode_{Curr} = M_{Curr} \mid Mode_A = M_A, Mode_B = M_B, Mode_D = M_D), \end{aligned} \tag{1}$$

where $Mode_{Curr}$, $Mode_A$, $Mode_B$ and $Mode_D$ are random variables that represent the RD optimal Intra_4x4 prediction modes of the current block and three neighboring blocks *A*, *B*, and *D* as depicted in Fig.3. M_{Curr}, M_A, M_B and M_D are their possible values respectively. It is by definition that $M_{Curr} \in \{0, 1, \ldots, 8\}$. To tackle the boundary case, we let M_A, M_B, and $M_D \in \{-1, 0, \ldots, 8\}$. Take block *A* as an example. If it does not exist, then $M_A = -1$. If the macro-block containing block *A* is not coded in Intra_4x4 mode, then $M_A = 2$, *i.e.*, the DC mode. The triple (M_A, M_B, M_D) is defined as the neighboring context.

Based on laws of large numbers, we can estimate conditional probabilities in different neighboring contexts from statistical frequencies. Eight sequences in CIF format, *i.e.*, 'City', 'Crew', 'Foreman', 'Harbour', 'News', 'Mobile', 'Stefan' and 'Tempete' are included in the statistics. Table 1 reveals some statistical results in the context (0, 0, 0). After estimating conditional probabilities with average frequencies, we sort modes in probability descending order. These orders denoted by $\{M_n(M_A, M_B, M_D), n = 0, 1, \ldots, 8\}$ are stored in Order Table indexed by neighboring contexts. For example, the item (0, 0, 0) records {0, 5, 7, 1, 2, 4, 6, 3, 8} as implicated in Table 1. In addition, the corresponding probabilities are stored in Probability Table. Instead of precise values, integers $\{U_n(M_A, M_B, M_D), n = 0, 1, \ldots, 8\}$ are recorded as

$$U_n(M_A, M_B, M_D) = \lfloor S \cdot P(M_n(M_A, M_B, M_D) \mid (M_A, M_B, M_D)) \rfloor, \tag{2}$$

where *S* is defined as the constant stable factor, to avoid floating-point operations. Empirically, we set *S* to be 30% of the total number of the 4x4 blocks in a frame.

Table 1. Percentages of Intra_4x4 modes in context (0, 0, 0).

	0	1	2	3	4	5	6	7	8
City	79.4%	3.7%	2.9%	1.9%	2.4%	3.3%	2.3%	3.0%	1.2%
Foreman	74.8%	4.0%	2.7%	2.6%	3.2%	3.7%	3.9%	3.4%	1.6%
Harbour	80.8%	2.3%	2.8%	1.8%	1.7%	4.0%	1.2%	4.1%	1.2%
Average	80.4%	3.0%	2.5%	1.9%	2.1%	3.5%	2.0%	3.5%	1.2%

3.2. Context Adaptive Coding

Based on the Markov random field hypothesis, we build up a context adaptive coding scheme to code the Intra_4x4 prediction mode information.

Firstly, in a specific neighboring context, we use the position of the mode in the Order Table as the symbol word to be coded. The position is counted from 0 to 8. For example, if we intend to code mode 1 in the neighboring context (0, 0, 0), the symbol word to be coded will be 3.

Secondly, we predefine ten Huffman code tables with index number 0 – 9, which are stored in VLC Table. Fig. 4 denotes these tables in the form of Huffman trees. Let H_k represent the Huffman code table with index number *k*, $C_k(n)$ represent the codeword for the symbol word n in H_k, and $L_k(n)$ represent the codeword length of $C_k(n)$. For instance, it can be seen from Fig. 4 that $C_1(2) = 011$, and $L_1(2) = 3$. If H_k is utilized in the neighboring context (M_A, M_B, M_D), then the expected codeword length in this context can be formulated as

$$\mathrm{E}\{L_k \mid (M_A, M_B, M_D)\} = \sum_{i=0}^{8} L_k(i) \cdot P(M_i(M_A, M_B, M_D) \mid (M_A, M_B, M_D)), \tag{3}$$

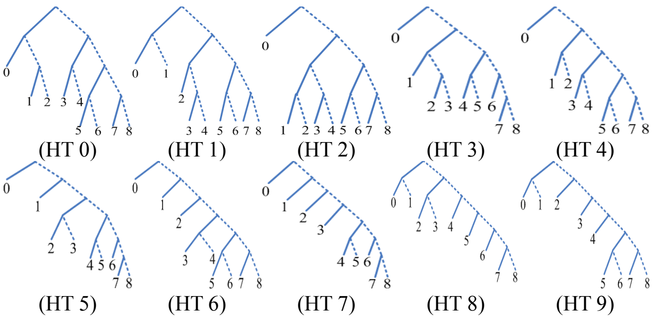
where L_k is a random variable denoting the codeword length with the Huffman code table H_k.

Thirdly, we assign an optimal Huffman code table to each neighboring context, so that the expected codeword length is the shortest. Formally, we define

$$T(M_A, M_B, M_D) = \arg\min_{k \in \{0,1,\ldots,9\}} \left\{\mathrm{E}\{L_k \mid (M_A, M_B, M_D)\}\right\}, \tag{4}$$

where $T(M_A, M_B, M_D)$ is the index number of the optimal Huffman code table in the context (M_A, M_B, M_D). Substituting (3) into (4), we can obtain the index numbers of the optimal code tables in all contexts. These index numbers will be stored into Lookup Table indexed by neighboring contexts.

Several steps are involved to code a mode M in the context (M_A, M_B, M_D). At first, the index number of the optimal code table, $k = T(M_A, M_B, M_D)$, is retrieved from Lookup Table. Thus the optimal code table H_k can be found from VLC table. Suppose mode M takes the nth position in Order Table, *i.e.*, $M = M_n(M_A, M_B, M_D)$, then n will be the symbol word to be coded. At last, $C_k(n)$ will be the codeword for mode M. It should be clarified that although we implement this coding approach in the CAVLC mode of H.264/AVC, it can also be adopted in the CABAC mode, as a binarization process.



(HT 0) (HT 1) (HT 2) (HT 3) (HT 4)

(HT 5) (HT 6) (HT 7) (HT 8) (HT 9)

Figure 4. Huffman Trees (HTs) used to code the Intra_4x4 prediction mode inforamtion. Solid lines and dashed lines represent 1s and 0s respectively. A path from the root node to a leaf node indicates a codeword.

Although the techniques described above work well, some redundancies still exist when coding the Intra_4x4 prediction mode information. In a specific neighboring context (M_A, M_B, M_D), we define the Intra_4x4 prediction mode taking the first place in Order Table, *i.e.*, $M_0(M_A, M_B, M_D)$, as the Most Probable Mode (MPM). In many cases, the probability that a 4x4 block holds its MPM is significantly large, as depicted in Table 1. As a result, in some macro-blocks with Intra_4x4 mode, almost all the 4x4 blocks are coded with MPMs. Apparently; it is inefficient to code the MPMs in these macro-blocks.

An implicit mode, called Intra_4x4_Implicity, is derived from Intra_4x4 mode to tackle this problem. Firstly, an additional bit is utilized in an Intra_4x4 macro-block to indicate whether it is coded in the implicit mode or not. Secondly, in a macro-block with this mode, each 4x4 block is coded using its MPM in the specific neighboring context, without conveying any bits for the Intra_4x4 prediction mode information. Thirdly, the RD cost of Intra_4x4_Implicity mode is calculated [6]. Then the new mode will be treated as one of the candidate macro-block modes in the RDO procedure.

3.3. Automatic Adjustments

Automatic adjustments play a significant role in the robustness of the proposed coding scheme. As mentioned above, our coding scheme is initialized with data learned off-line on a training set. Since the training set only consists of a limited number of sequences in CIF format, the initial state of the coding scheme may be not suitable for some sequences outside the training set or in different resolution. With automatic adjustments, the proposed algorithm can track individual characters of a variety of sequences.

After mode $M_t(M_A, M_B, M_D)$ is chosen to be the RD optimal one of the current 4x4 block, an updating process will refresh the tables. It should be noted that automatic adjustments only happen in the macro-blocks with Intra_4x4 mode, so that the decoder can be synchronous with the encoder. Thus the states of the tables should be restored if the final RD optimal mode of the current macro-block is not Intra_4x4 on the encoder side.

At first, Probability Table and Order Table are refined and reordered. $U_t(M_A, M_B, M_D)$ is updated as

$$U_t^{New}(M_A, M_B, M_D) = U_t^{Old}(M_A, M_B, M_D) + \delta\,, \tag{5}$$

where δ is defined as the convergence factor, and is chosen to be 5. The values of U_{t-1} (M_A, M_B, M_D) and U_t (M_A, M_B, M_D) will be switched, if

$$U_t^{New}(M_A, M_B, M_D) > U_{t-1}^{Old}(M_A, M_B, M_D) \text{ and } t > 0. \tag{6}$$

In such a case, the values of $M_{t-1}(M_A, M_B, M_D)$ and $M_t(M_A, M_B, M_D)$ in Order Table will also be exchanged.

After Probability Table and Order Table are refined, Lookup Table is adjusted. From (2), (3) is approximated as

$$\mathrm{E}\{L_k \mid (M_A, M_B, M_D)\} \doteq \frac{1}{S} \bullet \sum_{i=0}^{8} L_k(i) \cdot U_i(M_A, M_B, M_D)\,. \tag{7}$$

Then a new $T(M_A, M_B, M_D)$ can be obtained by utilizing (4), which will be stored into Lookup Table to replace the existing one. Actually, the calculation of the multiplication factor $1/S$ in (7) can be saved, since a common factor does not affect the result in (4).

It should be noted that the states of tables are re-initialized before coding each I-frame, since the coding procedure of an I-frame should not depend on the coding results of previous frames.

4. EXPERIMENTAL RESULTS

The proposed coding method is implemented in JM12.2 [8] and the original JM codec is treated as an anchor. Since we focus on the performance of intra coding, experiments are carried out for all I-frames sequences. H.264/AVC baseline profile is chosen for the sake of simplification. Sequences are tested in CIF or 4CIF format with QPs 28, 32, 36 and 40.

Firstly, we compare the composition of bits. Table 2 demonstrates the results, where ‘A’ represents the anchor and ‘P’ represents the proposed method. It can be seen that the bits used for Intra_4x4 prediction mode information in the proposed method are less than those in the anchor, especially at low bit-rates. Take the sequence ‘Foreman’ as an example. The percentage of those bits is reduced from 27.0% to 20.1% when QP = 40. This implicates that the proposed method compresses the Intra_4x4 prediction mode information more efficiently.

Secondly, we compare the composition of macro-block modes. Table 2 shows that Intra_4x4 mode, including the implicit one, takes a larger percentage in the proposed method than that in the anchor. Still take the sequence 'Foreman' as an example. The percentage of Intra_4x4 mode increases from 48.3% to 57.6% when QP = 40. Apparently, with the proposed mode coding method, Intra_4x4 mode becomes more competitive. Thus more macro-blocks prefer Intra_4x4 mode to Intra_16x16 mode in the RDO procedure.

Finally, we compare the RD performances. In Table 2, the average differences in PSNRs (*ΔPSNR*) and in bit-rates (*ΔBits*) are calculated [9]. The average gain is about 0.25 dB, which means an average bit-rate reduction about 4.3%. In the sequence 'Carphone', a gain up to 0.32 dB is reported. Fig.5 shows us two RD curves.

It should be noted that the proposed method improves the coding performance of sequences that are outside the training set as well, such as 'Carphone' in CIF format, and even 'Soccer' in 4CIF format. These results indicate the efficiency of automatic adjustments and the robustness of the proposed coding scheme.

As a price of the improvement in coding performance, the proposed method brings an additional computational complexity accordingly. Since table indexing, VLC coding and table updating are all light-weighted tasks, the decoder does not suffers from serious computational overheads. On the encoder side, the computational complexity increases slightly because one more macro-block mode is involved in the RDO procedure. In addition, the memory required by the tables is about 30 kilobytes, which is acceptable in most codec systems nowadays.

5. CONCLUSION

In this paper, we propose an efficient coding method for the Intra_4x4 prediction mode information in H.264/AVC. We build up a context adaptive coding scheme based on a Markov random field hypothesis. By adopting automatic adjustments, the proposed method adapts well to various sequences. Experiments show that we can compress the mode information more efficiently so as to obtain a gain up to 0.3 dB in all I-frames coding.

6. REFERENCES

[1] ITU-T, "Recommendation H.264," http://www.itu.int/rec/T-REC-H.264-200503-I/en, Mar. 2005.

[2] X. Fan, Y. Lu, and W. Gao, "A Novel Coefficient Scanning Scheme for Directional Spatial Prediction-based Image Compression," International Conference on Multimedia and Expo, pp 557-560, Jul. 2003.

[3] P. Zhang, D. Zhao, S. Ma, Y. Lu, W. Gao, "Multiple Modes Intra-prediction in Intra Coding," International Conference on Multimedia and Expo, pp 419-422, Jun. 2004.

[4] J. Yang, B. Yin, Y. Sun and N. Zhang, "A Block-matching Based Intra Frame Prediction for H.264/AVC," International Conference on Multimedia and Expo, pp 705-708, Jul. 2006.

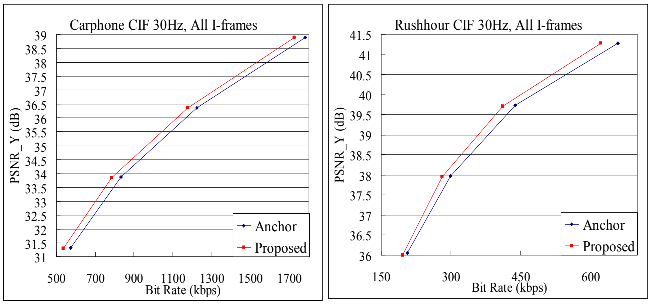
[5] T. Shiodera, A. Tanizawa and T. Chujoh, "Simulation Results of Bidirectional Intra Prediction on KTA Software Version 1.3," VCEG, Q6-AF06, San Jose, CA, Apr. 2007.

[6] G. J. Sullivan and T. Wiegand, "Rate-Distortion Optimization for Video Compression," IEEE Signal Processing Magazine, vol. 15, pp. 74-90, Nov. 1998.

[7] K. Zhang, Q. Wang, Q. Huang, D. Zhao, W. Gao, "A Context-based Adaptive Fast Intra_4x4 Prediction Mode Decision Algorithm for H.264/AVC Video Coding," PCS, Nov.2007.

[8] Joint Video Team (JVT), reference software JM 12.2, http://iphome.hhi.de/suehring/tml/download/jm12.2.zip.

[9] G. Bjontegaard, "Calculation of Average PSNR Differences between RD-Curves," VCEG, Q6-M33, Austin, , Apr. 2001.



(a) (b)

Figure 5. RD performance comparisons.

Table 2. Experimental results.

QP Seq.		Bit-streams Composition												Macro-blocks Composition								Performance	
		I4 Texture (%)				I4 Mode (%)				Others (%)				I16 (%)				I4 (%)				ΔPSNR (dB)	ΔBits (%)
		28	32	36	40	28	32	36	40	28	32	36	40	28	32	36	40	28	32	36	40		
Carphone_cif (250 frames)	A	68.3	61.8	53.2	41.5	16.2	19.6	22.6	23.9	15.5	18.6	24.3	34.6	28.9	37.5	47.9	59.4	71.1	62.5	52.1	40.6	0.32	-4.6
	P	72.4	67.4	61.3	52.5	13.3	16.1	18.0	18.6	14.2	16.5	20.7	28.9	24.2	31.2	39.9	49.7	75.8	68.8	60.1	50.3		
Foreman_cif (300 frames)	A	71.8	64.6	54.2	40.8	15.1	19.6	24.3	27.0	13.1	15.8	21.4	32.2	18.5	27.2	38.3	51.7	81.5	72.8	61.7	48.3	0.26	-3.5
	P	75.0	69.7	62.0	52.3	13.0	16.3	19.5	20.1	12.0	14.0	18.6	27.6	14.9	21.4	31.2	42.4	85.1	78.6	68.8	57.6		
Hall _cif (300 frames)	A	69.6	66.3	60.3	51.3	11.8	14.8	18.1	21.2	18.6	18.9	21.5	27.6	42.9	48.3	53.6	60.2	57.1	51.7	46.4	39.8	0.24	-4.2
	P	74.5	71.3	66.9	60.3	9.5	12.0	14.5	16.5	16.0	16.6	18.6	23.2	34.6	41.5	47.0	52.6	65.4	58.5	53.0	47.4		
Rushhour_cif (300 frames)	A	39.5	29.5	19.6	10.2	23.9	21.4	16.3	9.5	36.6	49.1	64.1	80.4	56.9	73.2	85.5	93.8	43.1	26.8	14.5	6.2	0.24	-5.3
	P	49.7	39.9	30.0	19.1	19.9	17.3	13.2	8.0	30.4	42.8	56.9	72.9	47.7	64.0	75.9	84.6	52.3	36.0	24.1	15.4		
Soccer_4cif (600 frames)	A	65.8	52.2	39.3	28.7	17.4	18.6	18.4	18.6	16.8	29.2	42.3	52.7	23.6	44.3	64.4	77.9	76.4	55.7	35.6	22.1	0.17	-3.8
	P	71.1	61.7	55.6	45.3	15.2	15.5	14.2	14.4	13.7	22.7	30.3	40.3	16.5	32.6	48.0	63.6	83.5	67.4	52.0	36.4		