EE-System, USC

EE669 Term Paper

Fast Mode Decision Based on Macroblock Motion Activity

Lindan Tian(3086874846)

5/3/2013

Term Paper of EE669

Fast Mode Decision Based on Macroblock Motion Activity

I. Motivation and Abstract

The motivation of the fast mode decision is that the initial mode decision part occupies a high portion of calculation time in the whole algorithm. According to [1], the exhaustive Mode Decision process takes up about 90% load of the total computational cost of the JM software. The Mode Decision in [1] includes the Motion Estimation part (to find the optimal motion vector for each mode) and the Mode determination part (to determine the best mode to encode). We have learned that the mode decided finally of a Macroblock should correspond to motion (fast or slow) and the texture (homogeneous or not). If we can get this temporal or spatial information of the current MB by some judgment conditions, it is appropriate to skip some less probable mode instead of calculating the RD cost of every possible mode exhaustively.

In this report, the MAMD (motion activity-based mode decision) algorithm proposed in [1] is first implemented strictly and then a little modification is applied to get better results. The reason of the modification and the analysis of the experimental results are also explained in detail.

II. Approach and Procedure

Instead of calculating the motion vector of each mode exhaustively and then compare the RDCost of all the modes, Zeng *et al.* [1] propose the MAMD algorithm to classify the Macroblock into one of the five classes correspond to five motion activity conditions before motion estimation. The first judgment for the classification is the RD cost of the skip mode (RD(Skip)). The classification details are showed in **Table 1.**

Table 1. The Motion Activity Classification [1]

Class	Motion Activity	Involved Modes
1	Motionless	SKIP
2	Slow Motion(homogeneous region)	SKIP 16X16
3	Moderate Motion	16x8,8x16
4	Fast Motion	P8x8
5	Highly-textured region in fast motion or with scene changes	I4MB, I16MB(Intra Mode)

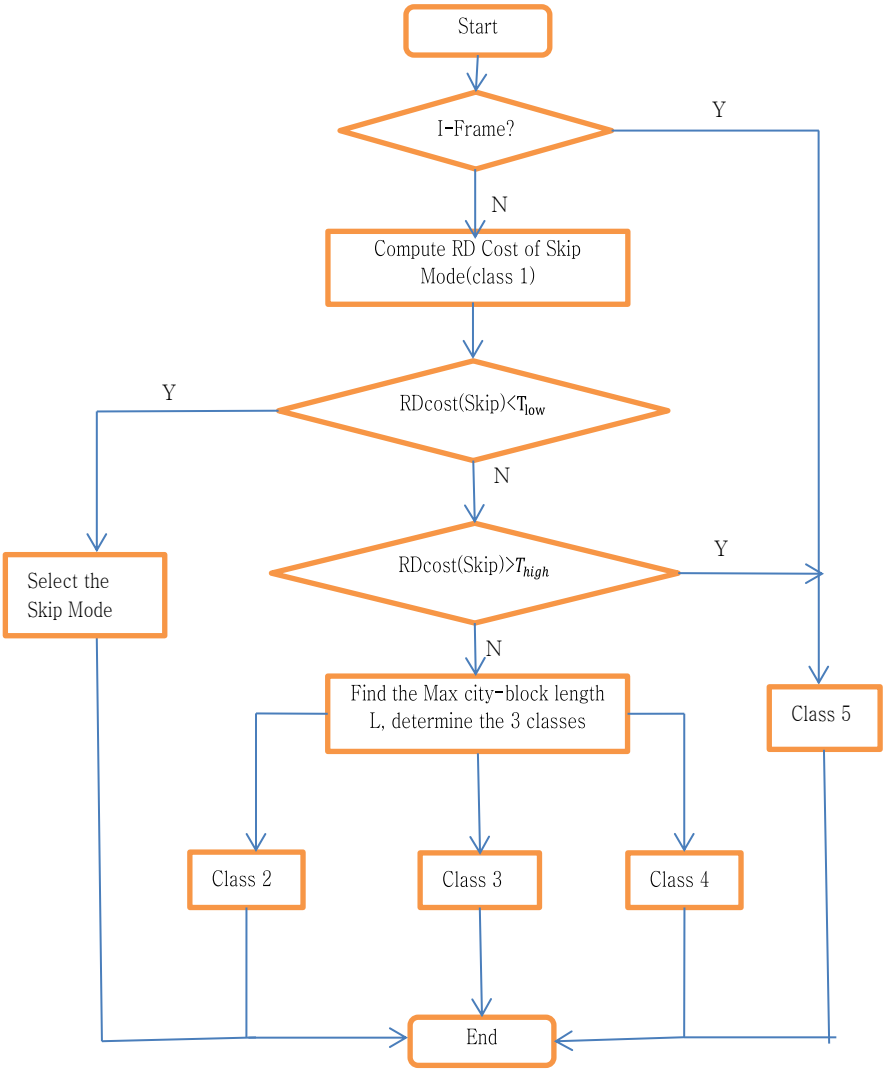


Fig.1. Flow Chart of MAMD Algorithm

The flow chart in Fig.1 illustrates the MAMD algorithm in detail based on the judgment conditions listed above.

At first step, we compute the RD cost of the Skip Mode and set the upper threshold T_{high} and lower threshold T_{low}. If the RDCost(Skip)< T_{low}. It implies that the motion of MB is small enough for us to just use the skip mode. If RDCost(Skip)> T_{high}, it implies that the MB may have a very high motion or even a scene change, which means we had better use the intra mode to do spatial prediction rather than the temporal prediction.

To set the value of the threshold, experiments are made in [1] and final formulas are obtained:

$$T_{low} = 34e^{0.1759QP} \quad (1)$$

$$T_{high} = 24215e^{0.675QP} \quad (2)$$

This final formula is obtained by training a list of videos and then applying curve fitting. In the implementation part, I used the Table 2 directly for the purpose of saving encoding time.

Table 2. Different Thresholds Corresponding to Different Qp Value

Qp	23	24	25	26	27	28
T_{low}	1700	2050	2500	3000	3650	4450
T_{high}	107500	116500	126500	137000	148500	160000
Qp	29	30	31	32	33	34
T_{low}	5400	6500	7850	9450	11400	13650
T_{high}	172000	185000	199000	214000	229500	245500
Qp	35	36	37	38	39	40
T_{low}	16350	19600	23250	27500	32500	38300
T_{high}	262000	279000	296500	314500	333000	351500

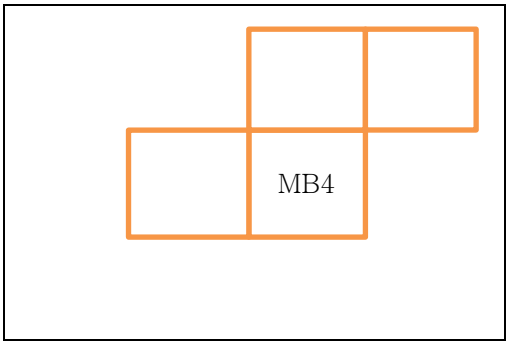
When the RDCost(Skip) is between the two boundary, we should make Mode Decision among the inter modes and 3 further classifications are applied. Here the value of "L" is used, which means the city block length [2]. We use four encoded MBs as the reference MB. In the current frame, we use the left, upper, upper right MBs as the references. In the previous frame, we use the collated MB as a reference. For one specific MB, the L(MB) means the sum of the MV's length of the x component and y component.

$$l_{vi} = |x_i| + |y_i| \quad (3)$$

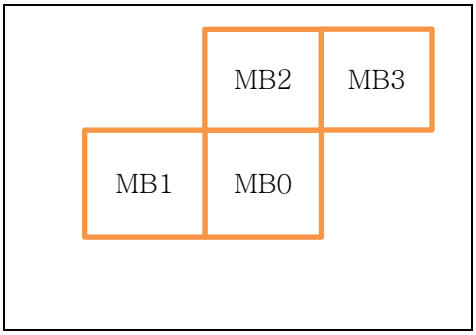
In the formula, v means motion vector, i is the index of the MB.

For each of the four reference MB, we can get one L value. Thus, the maximum city block length should be the maximum value of the all four Lengths. That is

$$L(MB0) = \max\{l_{vi}\}\ i = 1,2,3,4 \quad (4)$$



Previous Frame (Frame (n-1))



Current Frame (Frame n)

The l_{vi} means the MV of the whole MB, but we can know any of the four referenced MB may have been partitioned into smaller sub-blocks based on its final mode selected. Then each sub-

block may have its own MV. To get the MV for the whole MB, we can use the method proposed in [3], which is called as a bottom-up merging procedure. This method means we can get a block's MV by average the MVs of the further finer partitions in this block. Thus we can get one "MV" for each MB in the end.

In Jm18.4, the MVs are stored for each 4x4 block. I can find that the struct p_Vid->enc_picture->mv_info[][] stored the final optimal MVs for every 4x4 block for the current frame. Then using the merging idea in [3], I can simply add the 16 MVs in one MB together then divide the result by 16 without considering the mode and ref information. Another thing worthy notice is that the MVs stored in this struct is actually the 4 times of the real MV (for the design of the software to increase the "resolution" to enable probable Subpel ME). Thus, we need also divide the final result by 4.

Now the Maximum L has been obtained, we can use the value of the L to determine the classes.

$$Motion\ Activity = \begin{cases} Slow & L \le L_1 \\ Moderte & L_1 \le L \le L_2 \\ Fast & L_2 \le L \end{cases} \qquad (5)$$

$L_1 = 1$ and $L_2 = 2$ are used in [1]. The assumption hidden behind this judgment is that the information is highly related in spatial region (the three neighboring referenced MBs) and in temporal region (the use of the collated MB in the previous MB). We can use the MVs of the

four MBs to conclude the motion condition of the current MB. The bigger L means that the motion is bigger, which is indicated as “Fast” and should be partitioned with finer and smaller blocks, that is P8x8.

III. Analysis of the Algorithm and Discussion of Results and Modification

The first step of the algorithm to use the thresholds to classify the MB modes is very important. The Skip mode is to use the neighboring MBs to predict the motion of the current MB. In this way the RDCost(Skip) can reflect the homogeneous condition of the region motion. But we don’t need the T_{low} in fact: We have already calculated the RDCost(Skip), then we can compare all the modes in each classification with the performance of the Skip mode to get the mode with smaller RD cost. That means the class 1 and 2 can combine together and 3-5 can all add the skip mode without any extra calculation consuming.

At first, I implement the algorithm in [1] following each step strictly. However, the result I got is very bad compared to the result in [1]. Although the time saving is obvious, the bit rate increasing is too big, which means the RD Cost increases too fast. According to the statistics produced from running the encoding, the number of intra modes of the fast mode algorithm is far more than that of the original algorithm, which means the T_{high} may be too low to select the intra modes correctly. In this way, most of the MBs are classified into the intra modes, which

results in high residues (intra mode often leads to high residues). Then I tried to find the reason causing this bad performance. In the version of JM 18.4, The Distortion is first calculated as the SSE value of the current MB with the reference MB, then the SSE result is shifted 5 digits left (multiplied by 32) to increase the accuracy of the Motion Estimation to find the best motion vector. Resulting from the difference of the version of JM software (JM 10.2 used in [1]), the threshold may be not that powerful in my implementation. Then I tried to shift 5 digits of the threshold and got a better result. But we can know from [1] that the relation is not linear, which explains why the increasing of quality is very limited. And it is hard to find a robust enough threshold by simply trying.

The main goal of the fast mode decision is try to decrease the encoding time without severely affecting the RD Cost performance. The most time consuming part is actually the ME part. For each mode, the exhaustive search within the search range is carried out. In my experiment, the search range is 16, which means (32+1)*(32+1) =1089 RD cost evaluation will be conducted in the ME for selecting the best MV for each mode. That means the inter search takes up more load. Considering that the threshold used in [1] don't apply to our software that effectively and the relatively small computational load of the intra mode, I tried to abandon T_{high} and add all the intra modes (I4MB and I16MB) to class 2-4.

We assume that the city block length judgment for the inter modes is accurate enough, which means the L_1 and L_2 can classify the inter search modes without error. Then if I delete the class 5 and add intra modes to class 2-4, the resulting performance should be at least as good as the algorithm in [1] or even better. However, when I tried this method, although the quality after encoding has improved a lot even without little more time consuming, the performance is still bad compared to that of [1]. In [1], the bit rate increasing can be controlled in the region less than 1%, while my algorithm can only control the most increasing of bit rate less than 10%. The PSNR decreasing and time saving of my algorithm and [1] don't differ that much.

This can prove that the city block length judgment is not very accurate. The encoding experiment is on IPPP form of 300 frames with RDO on and 1 reference frame (same as the setting in [1]). The inaccurate coding frame will influence the following frames accumulatively (e.g. more residues accumulatively) since the H.264 uses the decoded previous frames as the reference. The [1] first excludes all the intra modes by the T_{high} in the classification of the inter modes. So for my algorithm, if one MB happened to be classified into the wrong class (2-4), then it should be compare with the intra mode, it is likely the intra mode has less RD cost compared to that of the wrong inter modes, which will lead to a worse result than [1] and errors accumulated greatly in

the 300 frames. We can find that the number of intra modes in my algorithm in bad performance is much more than in the original JM18.4 encoding, which explains the worse result.

Actually, I found it is no need to add the inter modes in all the 3 inter classification. The city block length reflects the motion condition of the MB. Thus we can only add the intra modes to the class 4, which means the high motion. However, we should notice that the "MV" for intra mode is set to 0. Actually the [1] implement in this way, the influence in this should be very small assuming the intra modes have correctly excluded in the first step and the L is tried to find the maximum value of l_i (e.g. If the single MB4 is intra mode, it will not affect) . But I cannot neglect this condition in my algorithm .Then considering the extreme case, all the 4 MB1-MB4 are intra modes, but we will classify this MB into smooth region, which is not reasonable. I have experimented different conditions to add the intra modes to the 3 classes and comparing the trade-off between the performance and the time saving. And I finally find the best method is to add the intra mode decision to class 2 and 4, deleted class 1 and 5. This corresponds to my analysis result above.

It can be observed my algorithm is worse than that of [1] no matter how I tuned the parameters and modify the algorithm. The probable reasons causing this are listed below:

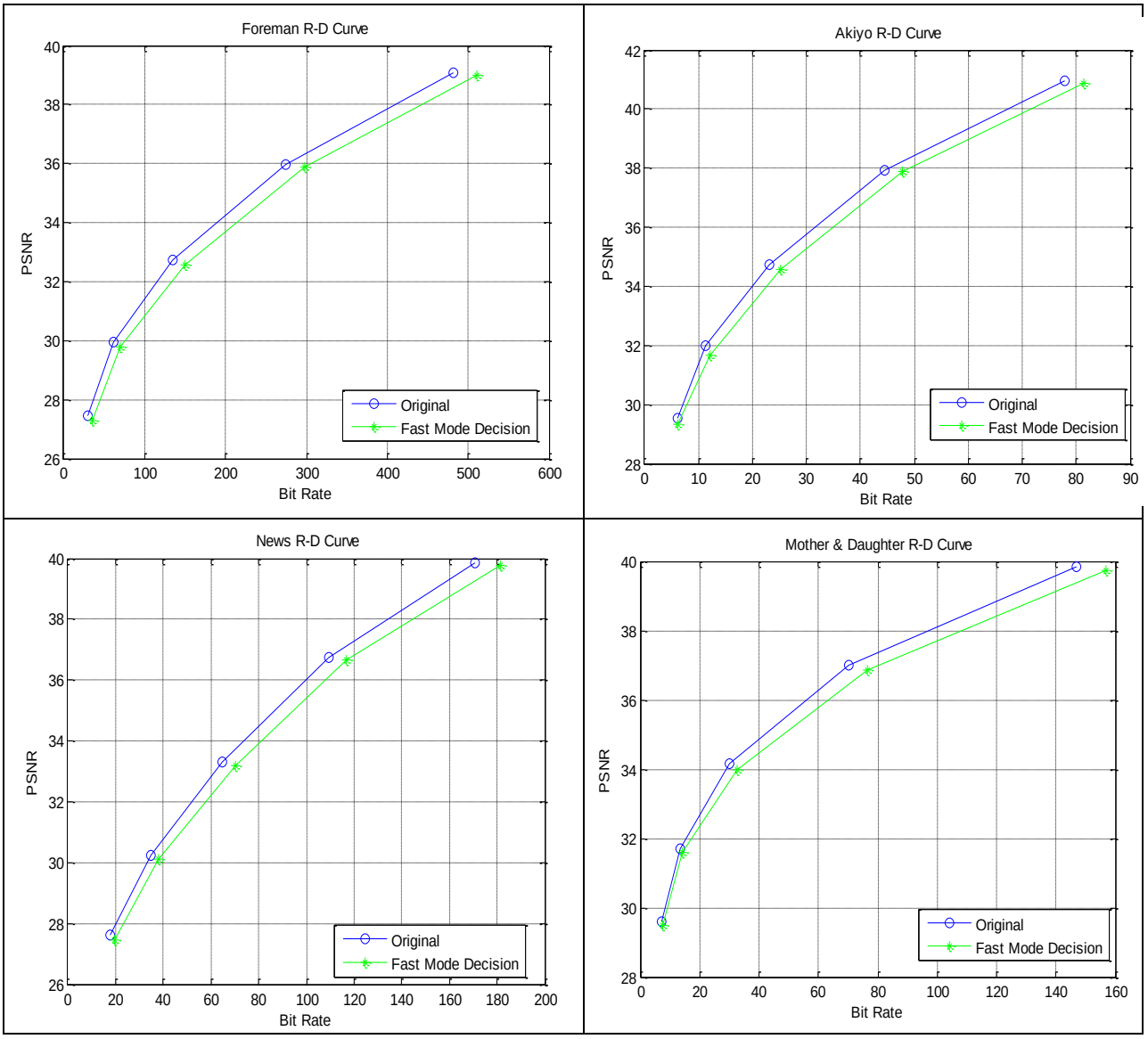
1) The T_{high} threshold is very strong and important and the city block length judgment is not that effective compared to T_{high}. I neglect the more effective factors in algorithm.
2) [1] listed a little configure setting parameters, I change these as [1] and left the other as default. But I disable the SubpelME to save the running time. I thought that the SubpelME doesn't affect the result that much because it is only a local refinement. But after attending the EE669 lecture of May 3th, I doubted that I may do something wrong, the residues can be much bigger without Subpel ME, let along the accumulation errors of 300 frames. I set the search range to be 16, while the information of the search range in [1] cannot be found. Although the configure set is same in the original JM18.4 and my implementation, I should notice that I skip many modes, which means the search range and the subpel may influence the results more.
3) The [1] decided the parameters T_{high}, T_{low}, L_1, L_2 using a large training set. However, it can be observed that the training set and the testing set are highly overlapped, which seem to be unreasonable. The value of L_1 and L_2 are obtained by training the format of both CIF and QCIF. The CIF and QCIF has different resolution, it seems to be unreasonable to use the same standard to evaluate the motion activity. In the testing part of [1] and my experiment, all the frames are in QCIF format, which may imply that the

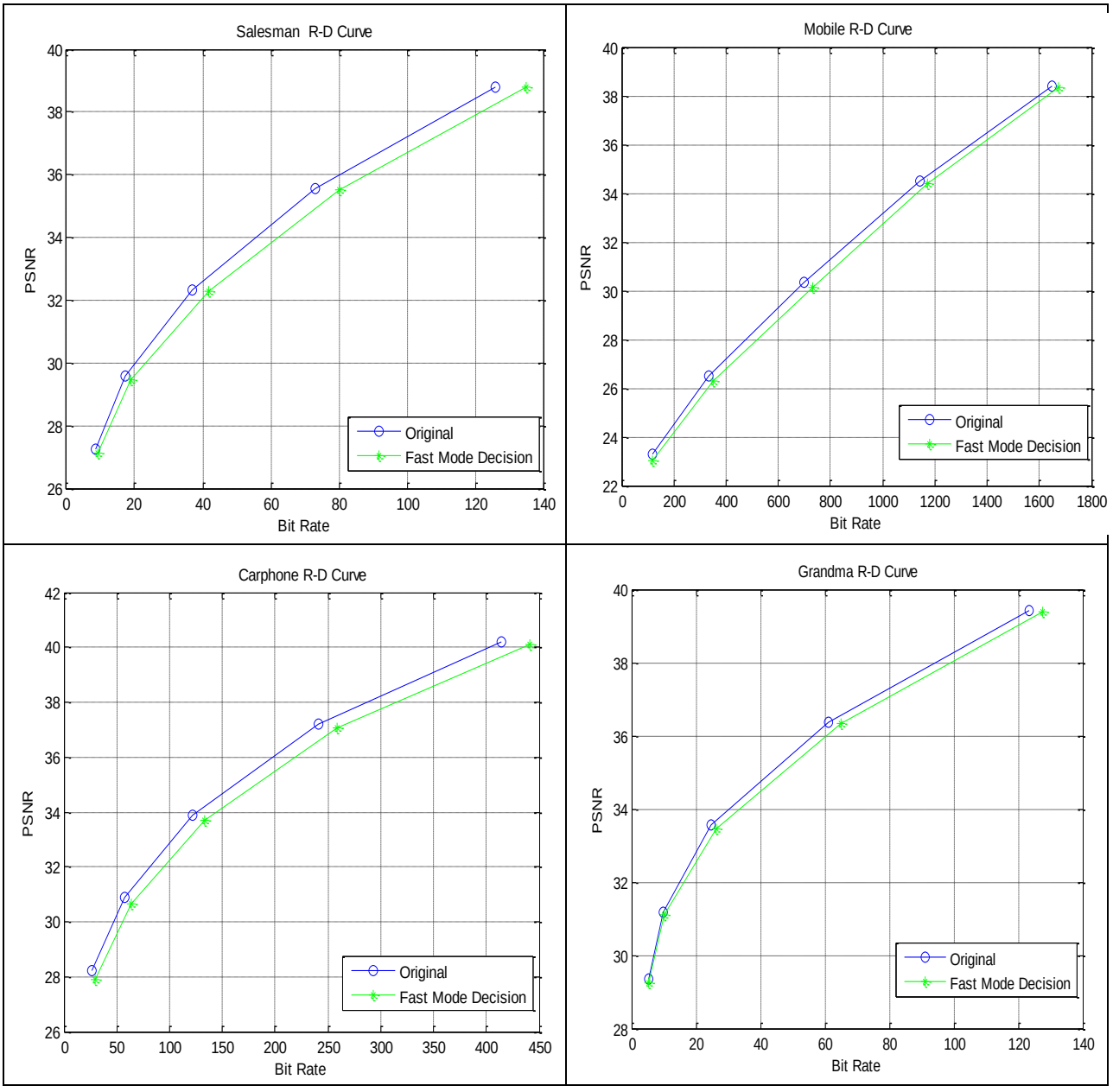
classification based on L_1 and L_2 are not accurate enough. This explains that the city block measure is not very strict.

We can also finde from the results that the videos with high bit rate in the initial JM encoding may get better results in my experiment. The example is the Mobile_qcif.yuv. This video has frequently scene changes which should result in more intra modes. Then the negative influence of the not strong city block length judgment can be relatively small. This corresponds to my experimental results: more intra modes.

IV. Experimental Result:

QCIF	Qp	PSNR(Fast)	Bit Rate(Fast)	Time(Fast)	PSNR(Original)	Bitrate(original)	Time(original)	ΔPSNR(dB)	ΔB(%)	ΔT(%)
Foreman	24	39.006	511.13	28.863	39.064	481.45	63.584	-0.058	6.16	-54.61
	28	35.883	297.39	26.212	35.973	274.16	57.365	-0.09	8.47	-54.31
	32	32.589	149.94	23.812	32.736	134.36	52.573	-0.147	11.6	-54.71
	36	29.772	69.83	21.294	29.935	61.9	48.277	-0.163	12.8	-55.89
	40	27.281	34.75	19.377	27.453	30.19	45.265	-0.172	15.1	-57.19
Akiyo	24	40.864	81.4	14.18	40.94	77.9	42.875	-0.076	4.49	-66.93
	28	37.878	47.79	13.644	37.92	44.53	40.986	-0.042	7.32	-66.71
	32	34.557	25.13	13.8	34.745	23.1	40.167	-0.188	8.79	-65.64
	36	31.68	12.03	13.715	31.985	11.28	38.105	-0.305	6.65	-64.01
	40	29.346	6.14	13.988	29.515	6.05	36.99	-0.169	1.49	-62.18
News	24	39.775	181.38	17.229	39.856	170.62	49.533	-0.081	6.31	-65.22
	28	36.647	117	16.589	36.733	109.26	46.491	-0.086	7.08	-64.32
	32	33.186	70.15	15.953	33.309	64.91	44.964	-0.123	8.07	-64.52
	36	30.128	37.89	15.427	30.223	34.64	42.931	-0.095	9.38	-64.07
	40	27.46	19.59	14.873	27.614	17.85	40.196	-0.154	9.75	-63
Mother	24	39.735	156.72	15.914	39.835	146.88	48.865	-0.1	6.7	-67.43
	28	36.852	76.59	15.077	36.988	70.24	45.255	-0.136	9.04	-66.68
	32	33.976	32.49	14.454	34.162	29.97	42.804	-0.186	8.41	-66.23
	36	31.59	14.26	14.322	31.707	13.54	40.733	-0.117	5.32	-64.84
	40	29.496	7.55	14.055	29.586	7.27	38.407	-0.09	3.85	-63.41
Salesman	24	38.784	135.12	16.532	38.789	125.87	50.519	-0.005	7.35	-67.28
	28	35.53	80.26	15.778	35.561	72.96	48.627	-0.031	10	-67.55
	32	32.295	41.62	15.415	32.328	36.81	47.815	-0.033	13.1	-67.76
	36	29.446	19.02	15.149	29.567	17.19	46.674	-0.121	10.6	-67.54
	40	27.142	9.08	15.472	27.265	8.62	45.035	-0.123	5.34	-65.64
Mobile	24	38.346	1675.17	24.545	38.418	1648.24	84	-0.072	1.63	-70.78
	28	34.395	1171.11	22.382	34.51	1143.71	77.797	-0.115	2.4	-71.23
	32	30.157	731.35	20.357	30.344	700.64	71.562	-0.187	4.38	-71.55
	36	26.286	349.29	18.756	26.511	334.65	64.557	-0.225	4.37	-70.95
	40	23.035	119.91	17.283	23.266	117.38	58.31	-0.231	2.16	-70.36
Grandma	24	39.371	127.72	15.896	39.425	123.27	50.068	-0.054	3.61	-68.25
	28	36.331	65.16	15.255	36.362	61.16	47.506	-0.031	6.54	-67.89
	32	33.466	26.45	14.46	33.549	24.69	44.87	-0.083	7.13	-67.77
	36	31.092	10.11	14.208	31.173	9.93	42.802	-0.081	1.81	-66.81
	40	29.251	5.33	14.063	29.338	5.35	39.039	-0.087	-0.4	-63.98
Carphone	24	40.094	441.14	24.975	40.191	414.68	57.733	-0.097	6.38	-56.74
	28	37.05	259.1	22.446	37.185	241.33	52.097	-0.135	7.36	-56.91
	32	33.714	132.98	20.506	33.864	122.28	47.555	-0.15	8.75	-56.88
	36	30.65	63.45	18.385	30.88	57.15	44.006	-0.23	11	-58.22
	40	27.905	29.32	16.785	28.244	26.49	41.712	-0.339	10.7	-59.76


Foreman R-D Curve
PSNR
Bit Rate
Original
Fast Mode Decision
Akiyo R-D Curve
PSNR
Bit Rate
Original
Fast Mode Decision
News R-D Curve
PSNR
Bit Rate
Original
Fast Mode Decision
Mother & Daughter R-D Curve
PSNR
Bit Rate
Original
Fast Mode Decision


Salesman R-D Curve
PSNR
Bit Rate
Original
Fast Mode Decision
Mobile R-D Curve
PSNR
Bit Rate
Original
Fast Mode Decision
Carphone R-D Curve
PSNR
Bit Rate
Original
Fast Mode Decision
Grandma R-D Curve
PSNR
Bit Rate
Original
Fast Mode Decision

V. Reference

[1] Huanqiang Zeng, Canhui Cai and Kai-Kuang Ma. Fast mode decision for H.264/AVC based on macroblock motion activity. *Circuits and Systems for Video Technology, IEEE Transactions on 19(4),* pp. 491-499. 2009.

[2] Kai-Chung Hou, Mei-Juan Chen and Ching-Ting Hsu. Fast motion estimation by motion vector merging procedure for H. 264. Presented at Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on. 2005 .

[3] P. I. Hosur and K.-K. Ma, "Motion vector field adaptive fast motion estimation," in Proceedings of the Second International Conference on Information, Communications and Signal Processing (ICICS), Singapore, Dec. 1999, pp. 234–246.