Random Initial Search Points Prediction for Content Aware Motion Estimation in H.264

Vidya N. More, Ajinkya Deshmukh, Dhiraj More and M. S. Sutaone
College of Engineering Pune, Wellesly Road, Shivaji Nagar, Pune, India

Keywords: H.264, Content Aware Motion Estimation, Search Point Prediction.

Abstract: Motion estimation algorithms used in video encoders are based on three important issues: selection of good initial search points, choice of appropriate search pattern and effective early termination criteria at different stages in algorithm.Motion vector prediction is also treated as initial search point prediction, in which possibility of good match block is predicted. Prediction is based on prior data from co-located and/or adjacent macroblocks from reference frame or current video frame respectively.Different search patterns contribute in achieving near accurate motion estimation. Different types of motion in real time videos can be tracked using different types of patterns. Early termination criteria at different stages in algorithm, avoid search at further possible locations which are pre-decided by pattern of search. This in turns reduces computations and motion estimation time.Proposed algorithm is combination of two concepts, content awareness and initial point prediction.Contents of video data is in terms of homogeneity coefficients. Initial search point prediction is used to avoid the search trapping into local minima.The algorithm is implemented on Reference Software of JM18.4 of H.264/AVC revised on 5th May 2011. The results of the implemented algorithm show that the total time taken for encoding and motion estimation time are less as compared with other algorithms for the videos of different resolutions.

1 INTRODUCTION

Video technology has been an inevitable and ubiquitous part of our daily lives. Right from mobile till the satellite surveillance systems, wide range of applications of the video technology have been implemented. The need for factors like higher quality, low bit rate and lesser disk space has triggered the development in this field. Even the hand-held devices now are able to play the high definition videos due to efficient encoding and decoding algorithms on the real time platforms.

Presently, H.264/MPEG-4 v10 is used widely as a standard video codec for all applications. H.264 applications mainly include satellite HDTV , high capacity storage devices such as Blu-ray discs, internet protocol television, video over internet, etc. The video coding standards are mainly divided into two main classes, namely, MPEG-x and H.26x. The MPEG-x codecs are developed ISO/IEC JTC1, whereas the H.26x are developed by ITU-T. However their joint work has resulted into standards such as H.262/MPEG-2 and H.264/MPEG-4 part 10. Current video standard H.265 is also published in 2013 which aims at higher throughput in all respects of video compression but at the cost of complicated algorithms and more hardware demand.It is observed that the motion estimation time takes more than 60% out of total time (refer to Table 1)thus there is need of optimum algorithms.Distortion measurement criteria, Sum of Absolute Difference (SAD)is mostly used in all versions of JM code. Table 1 show results of Total Time , ME Time , PSNR of Y and Bit Rate for a video sequence Foreman.yuv of qcif resolution on Simplified Unsymmetrical Hexagonal Search algorithm.

Motion estimation (ME)algorithms are quantified on the parameters of Total encoding time, ME time , Peak signal to noise ratio (PSNR) and Bitrate increase/decrease. These parameters are mainly outcome of three strategies of ME algorithm i.e. initial search point prediction, different search patterns of searching and early termination. Ample literature is available on search pattern selection and corresponding early termination threshold calculations.In this proposed algorithm we are focusing on initial point search on random basis and homogeneity analysis of video contents.

Heuristic search algorithm of ME is full search.

More V., Deshmukh A., More D. and Sutaone M..
Random Initial Search Points Prediction for Content Aware Motion Estimation in H.264.
DOI: 10.5220/0005311401590165
In *Proceedings of the 10th International Conference on Computer Vision Theory and Applications* (VISAPP-2015), pages 159-165
ISBN: 978-989-758-089-5

Table 1: More than 50% Time Taken by ME Compared with Full Search.

	ME Algorithm		
	Full Search	UMHEX	UMHEXS
Total Time	1142.068	318.435	293.341
ME Time	995.574	169.757	146.809
PSNR	35.805	35.76	35.78
Bitrate	1115.5	1168.99	1123.43

It searches the macroblock for each possibility of match in reference video frame and thus it is the optimum algorithm.Different block based ME algorithms are developed to minimize the search points in search range so as to reduce the ME Time. The algorithms like Three step search (TSS), New three step search (NTSS) (Renxiang Li and Liou, 1994), Four step search (FSS)(Po and Ma, 1996), Diamond search (DS) (Zhu and Ma, 1997),Hexagonal search (Zhu et al., 2002), Unsymmetrical Hexagonal search (UMHEX)(Zhibo Chen, 2002) ,Simplified UMHEX(UMHEXS) (Toivonen and Heikkil, 2006), Adaptive Rood Pattern Search(Nie and Ma, 2002) Predicted Motion Vector Field Adaptive Search Technique(PMVFAST) (Tourapis et al., 2002) and Enhanced Predictive Zonal Search (EPZS)are developed (Xu and He, 2008). Among these algorithms TSS,NTSS,FSS were part of MPEG 1,Mpeg 2 and UMHEX, PMVFAST and EPZS are part of current video standards(Sarwer and Wu, 2009). Thus it will be more appropriate to compare the results with these algorithms.All these mentioned algorithms are mainly focused on different shapes of search patterns and criteria to select these patterns(K. Venkatachalapathy and Viswanath, 2004). Figure 1 show some of the search patterns taken by UMHEXS algorithm at different stages in the algorithm.

Researchers are developing the algorithms for content aware ME to make the search more adaptive (Yi-Ching Liawa and Zuu-ChangHong, 2009). Characteristics of video data can be in terms of object acceleration, inertia, linearity (DongYoon Kim and Park, 2013) of the object etc. This paper is focusing on Homogeneity of the object in a macroblock with respect to adjacent macroblocks of the video frame. The algorithm developed by Humaria Nisar is implemented for the proposed work as base work. In proposed algorithm initial search points are generated randomly which are used in tracking the non-homogeneous/irregular motion of video data. Thus this is an incremental work in the base algorithm .
This paper has five sections and from which section 2 details about basic algorithm of homogeneity.Proposed work is mentioned in section 3 along with Simulation and test conditions is discussed in section 4. Results and conclusions are elaborated in section 5 and the references are listed used for this work.

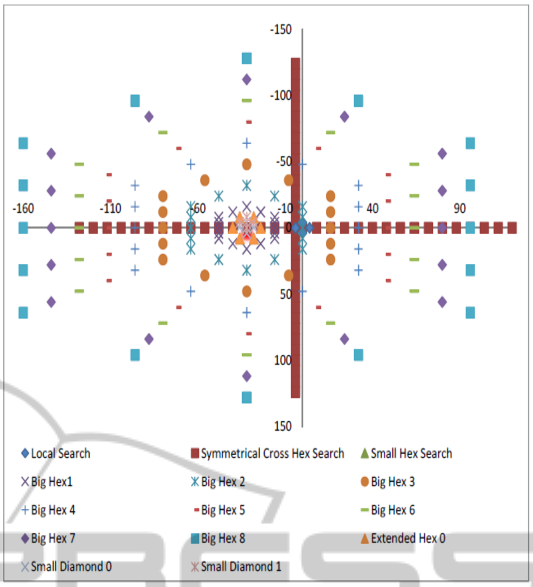

Figure 1: Example of search patterns in UMHEXS algorithm.

2 HOMOGENEITY ALGORITHM

Every video sequence contains varying amount of homogeneity.i.e.various macroblocks of frames of these sequences belong to one group or another. Thus making it easy for us to predict the motion vector with the help of previously coded homogeneous blocks(Humaira Nisar and Choi, 2012). This algorithm classifies every macroblock into three categories :

- Homogeneous
- Non homogeneous
- Stationary

and depending upon these categories the search pattern is applied adaptively.

2.1 Motion Vector Prediction

Motion vector prediction is carried out using spatio temporal neighbours.The spatial neighboring blocks in current frame are left, right, corner left, corner right macroblocks, whereas the temporal domain provides the collocated macroblock which is the macroblock with the same location in the reference frame. These neighbors provide three kinds of initial motion vectors,

1. Zero motion vector(ZMV=[0,0])
2. PMV1=median(motion vectors of spatial neighbors)
3. PMV2=motion vector of collocated macroblock

2.2 Homogenity Analysis

In video sequences, there exists high correlation between the neighboring blocks in the spatial and temporal domains. If the current and neighboring blocks belong to same object then they have consistent motion activity and hence these can be classified as homogeneous blocks. If the motion vectors are not consistent then these blocks are considered as non homogeneous blocks. If the blocks are homogeneous then a simple median prediction is carried out. The homogeneity coefficients (HC) play important role in the analysis.
Calculate the average of motion vectors of neighboring blocks in the current frame.

$$\overline{MV}_x = \left(\frac{1}{N}\right)\sum_{i=1}^{N} MV_{xi} \quad (1)$$

$$\overline{MV}_y = \left(\frac{1}{N}\right)\sum_{i=1}^{N} MV_{yi} \quad (2)$$

Here N is number of neighboring blocks of current block (In this case N = 5 i.e. A, B, C, D blocks from same frame and co-located block from reference frame).
The homogeneity coefficients are

$$HC_x = \frac{\left(|\sum_{i=1}^{N} \overline{MV}_{xi} - MV_{xi}|\right)}{|\overline{MV}_x|} \quad (3)$$

$$HC_y = \frac{\left(|\sum_{i=1}^{N} \overline{MV}_{yi} - MV_{xi}|\right)}{|\overline{MV}_y|} \quad (4)$$

$$HC = HC_x + HC_y \quad (5)$$

However, for $\overline{MV_x}$=0 or $\overline{MV_y}$=0, there are two possible cases.

1. The x or y components of motion vectors of neighboring blocks lie in opposite directions to each other.
2. The neighboring blocks are stationary i.e. all neighboring blocks have coefficients (0,0).

In such cases HC are calculated using the mean of absolute values of x and y components of the *MVs*.

If MV_x=0 and MV_y=0, then that block is considered as a Stationary block.
Smaller the value of HC means the block is homogeneous block and motion is consistent. Large value of HC denotes that the block is non-homogeneous.

2.3 Predicted Motion Vector

The magnitude of predicted motion vector(PMV) provides the basis for motion classification and hence defines the motion content of the video. If the magnitude of the PMV is greater than 1/2 of the search range, motion of block is considered fast else it is medium or slow. The PMV is calculated using following equation,

$$PMV = median(MV_A, MV_B, MV_C, MV_D) \quad (6)$$

2.4 GMP Calculations

The SAD of the global minimum is generally small as compared to the neighboring values. Hence the error descent rate is quite sharp. The GMP can be calculated as,

$$GMP = \frac{SAD_{neighbourhood}}{SAD_{centre}} \quad (7)$$

where $SAD_{neighbourhood}$ is the SAD of the point next to the center point and SAD_{centre} is the SAD of the center point. The smaller the value of GMP, closer we are to the global minimum. If the value of GMP is larger then the search is far away from global minimum and we have to incorporate larger search pattern and multiple predictors to identify the direction of global minimum.

2.5 Early Termination

The early termination threshold is different for homogeneous and non-homogeneous blocks.For homogeneous blocks,

$$ET_H = mean(SAD_A; SADl_B; SAD_C; SAD_D) \quad (8)$$

For non-homogeneous blocks,

$$ET_{NH} = min(SAD_A; SADl_B; SAD_C; SAD_D) \quad (9)$$

3 PROPOSED ALGORITHM

The algorithm described above distinguishes between slow and fast motion of video sequence. But the performance of the algorithm can be improved in case of fast and irregular motion video sequences. The randomized prediction can be performed in order to cater this need. If the motion is fast and irregular then there is a higher possibility that the neighboring blocks may not be able to provide accurate motion vector predictors hampering the performance in terms of motion estimation time and PSNR. Thus randomized motion prediction can be performed in the specified search range and the performance can be lifted. The performance of any algorithm is measured in terms of,

- Total encoding
- ME time
- PSNR of the luma component
- Bit rate

All the above mentioned parameters are interdependent. However, if you try to reduce the encoding and estimation time, the PSNR decreases resulting in increased bitrate. On the other hand,if we try to increase the PSNR and decrease the bit rate, complexity increases causing increase in encoding and estimation time. Thus we can not use any one approach blindly for all the applications. The approach is chosen depending upon the application need.

Parameter	Value
GOP Structure	IPPPPPP
Profile IDC	Baseline(66)
Quantization Parameter	I=28,P=28
Search Range	32
Entropy Coding Method	CAVLC
Block distortion measure	SAD

3.1 Algorithm

As mentioned earlier, fast and irregular motion can be compensated with the help of random motion prediction.The steps followed in this approach are given below,

1. Calculate PMV of the macroblock (refer section 2.3).
2. If macroblock is either stationary or slow set prediction range 16X16.For fast macroblock set it to 32X32.
3. Divide assigned prediction range in 4 parts.
4. From each part , one coordinate is randomly generated as Alpha from upper left,Beta from upper right, Gamma from lower right and Delta from lower left.
5. Center of prediction range is called as Epsilon.
6. Calculate SAD at each of 5 points.
7. Compare this SAD and find point with minimum SAD.
8. If this minimum SAD is lower than SAD of Spatio-temporal neighbour then assign this randomly generated point as Search center.If SAD is not minimum find minimum SAD point among neighbour.
9. Apply pre-defined search pattern on selected point for further search.

4 IMPLEMENTATION AND TEST CONDITIONS

Working environment The proposed algorithm is implemented in JM reference software version 18.4 of H.264 standard. The implementation is carried out in Linux environment and the processor used is Intel core i7 processor.

Following test conditions were used by making changes in configuration file.
The test sequences used are taken from different data bases. The links are listed below,
http://trace.eas.asu.edu/yuv/
ftp://ftp.tnt.uni-hannover.de
ftp://ftp.ldv.e-technik.tu-muenchen.de
Test sequences are of different resolutions as shown below.

- Bus(QCIF)
- Clair(QCIF)
- Coastguard(QCIF)
- Container(QCIF)
- Carphone(QCIF)
- Mobile(QCIF)
- Bus(CIF)
- Mobile(CIF)
- Shields(SD)
- Ice (HD)

Since most of the videos have motion changes or scene changes after 150th frame, all these video sequences are executed for 300 frames. The algorithm is meant for generating random data points in four quadrants, it is iterated for 10 times for the same test conditions and the average values of total time, ME time, PSNR of Y and Bit Rate are presented in tabular form.
The performance of proposed algorithms is compared with three algorithms already present in the JM 18.4 reference software, viz.,

1. Fast full search (FFSearch)
2. UMHEX
3. UMHEXS

5 RESULTS AND CONCLUSION

Table 2 is representing the actual values of the test parameters where as table 3 represents normalized values of parameters.It is observed from the results that for all tested video sequences total encoding time and

Table 2: Results for Different Videos.

Video	Algorithm Name	Total Time (Sec)	ME Time (Sec)	PSNR (dB)	Bit Rate
bus_qcif	FFSearch	107.11	71.552	34.455	212.24
	Umhex	76.168	39.304	34.409	225.99
	Umhexsmp	68.477	32.435	34.424	214.49
	Homogenity	59	22.797	34.361	231.23
	Proposed	58.1507	21.6406	34.3655	243.61
carphone_qcif	FFSearch	1180.938	1064.747	36.789	150.71
	Umhex	262.952	146.67	36.729	153.85
	Umhexsmp	221.124	105.73	36.73	151.68
	Homogenity	203.334	88.935	36.711	153.88
	Proposed	203.157	87.4971	36.7138	154.677
claire_qcif	FFSearch	1526.664	1424.182	39.744	31.03
	Umhex	226.552	126.101	39.596	31.29
	Umhexsmp	167.9	68.321	39.533	30.94
	Homogenity	172.077	72.528	39.615	30.96
	Proposed	169.5722	68.9847	39.533	30.94
coastguard_qcif	FFSearch	969.892	848.188	34.211	170.13
	Umhex	273.97	150.728	34.204	170.99
	Umhexsmp	228.006	107.192	34.207	170.56
	Homogenity	192.873	72.631	34.185	170.81
	Proposed	191.5548	70.7902	34.1913	171.276
container_qcif	FFSearch	923.489	830.41	35.992	36.45
	Umhex	174.499	81.791	35.961	36.86
	Umhexsmp	144.53	51.832	35.941	36.58
	Homogenity	140.355	48.513	35.939	36.98
	Proposed	140.4631	48.1663	35.9345	36.659
foreman_qcif	FFSearch	928.134	830.147	36.06	125.09
	Umhex	231.246	132.687	35.993	129.7
	Umhexsmp	203.169	104.072	36.001	126.23
	Homogenity	175.967	78.738	36.011	130.96
	Proposed	173.8248	76.202	35.9895	132.266
mobile_qcif	FFSearch	963.292	777.915	33.22	378.69
	Umhex	331.908	143.395	33.201	379.99
	Umhexsmp	331.011	143.66	33.201	379.99
	Homogenity	266.264	71.902	33.207	380.8
	Proposed	255.4348	69.8024	33.2091	380.978
bus_cif	FFSearch	1995.345	1740.457	35.029	940.92
	Umhex	604.852	345.656	34.988	1005.84
	Umhexsmp	547.367	293.197	35.023	949.58
	Homogenity	444.73	188.965	34.966	1024.88
	Proposed	429.1194	174.3156	34.9576	1050.853
mobile_cif	FFSearch	3735.533	2942.544	34.12	1478.56
	Umhex	1231.955	570	34.085	1489.51
	Umhexsmp	1123.831	467.731	34.101	1482.85
	Homogenity	960.884	308.059	34.083	1502.78
	Proposed	954.6059	300.3608	34.0842	1499.15
shields_sd	FFSearch	25606.11	22937.532	35.828	923.46
	Umhex	6221.245	3441.304	35.777	937.41
	Umhexsmp	5452.145	2800.285	35.8	922.94
	Homogenity	4750.687	2119.94	35.782	931.61
	Proposed	4660.5152	2014.2926	35.7696	935.503
ice_hd	FFSearch	66966.356	64304.688	41.595	2167.16
	Umhex	7963.397	5263.388	41.508	2385.62
	Umhexsmp	5977.357	3365	41.54	2189.28
	Homogenity	5791.631	3199.114	41.534	2254.16
	Proposed	5690.359	3112.7386	41.535	2293.154

Table 3: Normalized Results for Different Videos.

Video	Algorithm Name	Total Time Saving (%)	MET Saving (%)	PSNR Loss(dB)	Bit Rate Increases(%)
bus_qcif	FFSearch	–	–	–	–
	Umhex	28.888059	45.06932021	0.133507474	-6.478514889
	Umhexsmp	36.06852768	54.6693314	0.089972428	-1.060120618
	Homogenity	44.91644104	68.13925537	0.27281962	-8.947418017
	Proposed	45.70936421	69.75542263	0.259759106	-14.78043724
carphone_qcif	FFSearch	–	–	–	–
	Umhex	77.73363208	86.22489662	0.163092229	-2.083471568
	Umhexsmp	81.27556231	90.0699415	0.160374025	-0.643620198
	Homogenity	82.78199194	91.64731152	0.212019897	-2.103377347
	Proposed	82.79698003	91.78235769	0.204408927	-2.632207551
claire_qcif	FFSearch	–	–	–	–
	Umhex	85.16032342	91.14572435	0.372383253	-0.837898808
	Umhexsmp	89.0021642	95.20279009	0.530897746	0.290041895
	Homogenity	88.7285611	94.90739245	0.324577295	0.225588141
	Proposed	88.89263125	95.1561879	0.530897746	0.290041895
coastguard_qcif	FFSearch	–	–	–	–
	Umhex	71.75252502	82.2294114	0.020461255	-0.505495797
	Umhexsmp	76.49160938	87.36223573	0.011692146	-0.252747899
	Homogenity	80.11397145	91.436922	0.075998948	-0.399694351
	Proposed	80.24988349	91.65394936	0.057583818	-0.673602539
container_qcif	FFSearch	–	–	–	–
	Umhex	81.10437699	90.15052805	0.086130251	-1.124828532
	Umhexsmp	84.34956995	93.75826399	0.141698155	-0.356652949
	Homogenity	84.80165979	94.15794607	0.147254946	-1.454046639
	Proposed	84.78995418	94.19969654	0.159757724	-0.573388203
foreman_qcif	FFSearch	–	–	–	–
	Umhex	75.08484766	84.01644528	0.185801442	-3.68534655
	Umhexsmp	78.10994964	87.46342515	0.163616195	-0.911343832
	Homogenity	81.04077644	90.51517382	0.135884637	-4.692621313
	Proposed	81.27158363	90.82066188	0.195507488	-5.736669598
mobile_qcif	FFSearch	–	–	–	–
	Umhex	65.54440398	81.56675215	0.057194461	-0.343288706
	Umhexsmp	65.63752216	81.53268673	0.057194461	-0.343288706
	Homogenity	72.35895243	90.75708786	0.039133052	-0.557183976
	Proposed	73.48313907	91.02698881	0.032811559	-0.604188122
bus_cif	FFSearch	–	–	–	–
	Umhex	69.68684613	80.13992877	0.117045876	-6.899630149
	Umhexsmp	72.56780156	83.15402219	0.017128665	-0.920375802
	Homogenity	77.7116238	89.14279411	0.179850981	-8.923181567
	Proposed	78.49397473	89.98449258	0.203831111	-11.68356502
mobile_cif	FFSearch	–	–	–	–
	Umhex	67.02063668	80.62900674	0.102579132	-0.740585434
	Umhexsmp	69.91510984	84.10453675	0.055685815	-0.29014717
	Homogenity	74.2771915	89.53086173	0.108440797	-1.638080294
	Proposed	74.44525587	89.79247889	0.104923798	-1.39257115
shields_sd	FFSearch	–	–	–	–
	Umhex	75.70406048	84.99706071	0.142346768	-1.510623091
	Umhexsmp	78.70764048	87.79169006	0.078151167	0.056309965
	Homogenity	81.44705697	90.75776766	0.128391202	-0.882550408
	Proposed	81.79920652	91.21835514	0.163001005	-1.304117125
ice_hd	FFSearch	–	–	–	–
	Umhex	88.10836146	91.81492335	0.209159755	-10.08047398
	Umhexsmp	91.0740895	94.76710003	0.132227431	-1.020690674
	Homogenity	91.35143175	95.02506878	0.146652242	-4.014470551
	Proposed	91.50265993	95.15939087	0.144248107	-5.813783938

ME time is saved. ME Time saved is ranging from 0.4% to 1.62%. Most of the videos considered as test sequences are having higher motion in their video contents thus it can be said that it is saving more time than the other algorithms. Also as the resolution of video is increased no substantial change is observed in Homogeneity algorithm and proposed algorithm in terms of time but bit rate is increasing by .04 to 0.08 %.

ACKNOWLEDGEMENTS

This work is supported by Center of Excellence - Signal and Image processing, College of Engineering, Pune,India under Technical Education Quality Improving Program (TEQIP) PHASE-II.

REFERENCES

DongYoon Kim, H. L. and Park, H. (2013). Iterative true motion estimation for motion-compensated frame interpolation. In *IEEE Transactions on Circuits and Systems for Video Technology, Vol. 23, NO. 3*. IEEE.

Humaira Nisar, A. S. M. and Choi, T.-S. (2012). Content adaptive fast motion estimation based on spatio-temporal homogeneity analysis and motion classification. In *Pattern Recognition Letters 33*. IEEE.

K. Venkatachalapathy, R. K. and Viswanath, K. (2004). A new adaptive search strategy for fast block based motion estimation algorithms. In *Journal of Visual Communication and Image Representation. 15*. Science Direct.

Nie, Y. and Ma, K.-K. (2002). Adaptive rood pattern search for fast block-matching motion estimation. In *IEEE Transactions on Image Processing, Vol. 11, NO. 12*. IEEE.

Po, L.-M. and Ma, W.-C. (1996). A novel four-step search algorithm for fast block motion estimation. In *IEEE Transactions on Circuits and Systems for Video Technology, Vol. 6, NO. 3*. IEEE.

Renxiang Li, B. Z. and Liou, M. L. (1994). A new three-step search algorithm for block motion estimation. In *IEEE Transactions on Circuits and Systems for Video Technology, Vol. 4, NO. 4*. IEEE.

Sarwer, M. G. and Wu, Q. J. (2009). Adaptive search area selection of variable block-size motion estimation of h.264/avc video coding standard. In *IEEE Transactions on Circuits and Systems for Video Technology, Vol. 18, N*. IEEE.

Toivonen, T. and Heikkil, J. (2006). Improved unsymmetric-cross multi-hexagon-grid search algorithm for fast block motion estimation. In *JInternational Conference on Image Processing (ICIP)*. IEEE.

Tourapis, A. M., Au, O. C., and Liou, M. L. (2002). Highly efficient predictive zonal algorithms for fast block-matching motion estimation. In *IEEE Transactions on Circuits and Systems for Video Technology, VOL. 12, NO. 10*. IEEE.

Xu, X. and He, Y. (2008). Improvements on fast motion estimation strategy for h.264/avc. In *IEEE Transactions on Circuits and Systems for Video Technology, Vol. 18, NO. 3*. IEEE.

Yi-Ching Liawa, J. b. and Zuu-ChangHong (2009). Fast block matching using prediction and rejection criteria. In *Journal on Signal Processing*. ELSEVIER.

Zhibo Chen, Peng Zhou, Y. H. (5-13 December, 2002). Fast integer pel and fractional pel motion estimation for jvt. In *Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6) 6th Meeting: Awaji, Island, JP*. ITU-T — ISO/IEC Joint Video Coding Experts Group (JVT) File:JVT-F017.doc,.

Zhu, C., Lin, X., and Chau, L.-P. (2002). Hexagon-based search pattern for fast block motion estimation. In *IEEE Transactions on Circuits and Systems for Video Technology, Vol. 12,NO.5*. IEEE.

Zhu, S. and Ma, K.-K. (1997). A new diamond search algorithm for fast block matching mot ion estimation. In *International Conference on Information, Communications and Signal Processing*. IEEE.