A Comparison of Block-Matching Motion Estimation Algorithms

María Santamaría and María Trujillo
Multimedia and Vision Laboratory
School of Computing and Systems Engineering, Universidad del Valle, Colombia
{maria.santamaria, maria.trujillo}@correounivalle.edu.co

***Abstract*—Block-matching motion estimation is an efficient algorithm for reducing the temporal redundancy in video coding and is adopted by video coding standards. Many fast block-matching algorithms have been devised to reduce the computational complexity without degrading the estimation quality. Seven algorithms are implemented and compared – using quality of prediction and efficiency: the full-search, the three-step search, the four-step search, the diamond search, the hexagonal block search, the multi-directional gradient descent search and the fast directional gradient descent search. The aforementioned algorithms are chosen not only because of their popularity, but also because they are rather generic and they represent different ways of cutting down the computation. The most efficient algorithms are the hexagonal block search, the multi-directional gradient descent search and the fast directional gradient descent search. The hexagonal block search produces low quality of prediction, while the two others have the best quality prediction among all analysed algorithms. Also, the hexagonal block search has shown to be less affected by the variation in the block size.**

***Index Terms*—Block-matching, motion vector estimation, video coding, video sequences.**

I. Introduction

The block-matching algorithm (BMA) is a standard technique for encoding motion in video sequences [1]. It is widely used in applications such as: video compression – for which it was thought initially –, stereo vision and object tracking. Typically, the current frame is divided into non-overlapping blocks and for each of block, the algorithm searches for the best matched block using a search window. The relative position between the reference block and its best matched block is represented as the motion vector [2].

Block-matching is perhaps the most reliable and robust technique for motion estimation in video coding. Different measures have been proposed to determine the match between two blocks. The two most popular are: the Mean Square Error (MSE) and the Sum of Absolute Differences (SAD). Let MxN be the size of two images X and Y, where X is the reference image and Y is the distorted image, the MSE and the SAD are calculated as follows:

$$MSE(X,Y) = \frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}|X(i,j)-Y(i,j)|^2 \quad (1)$$

$$SAD(X,Y) = \sum_{i=0}^{M-1}\sum_{j=0}^{N-1}|X(i,j)-Y(i,j)| \quad (2)$$

The MSE is more accurate than the SAD, in terms of quality [3]. However, the latter is more used because it involves less computational cost and also the SAD is closed to the MSE, in term of results. Moreover, block-matching algorithm is computational expensive. Different approaches have been proposed in order to improve block-matching motion estimation accuracy and efficiency. The more expensive operation is the computation of distortion – the match – for a couple of blocks. Thus, the speed of a BMA is a function of the number of explored blocks.

Motion vectors can be used to predict changes in the scene between two or more frames of a video. Thus, the data size is reduced to an encoding only the current frame of a video sequence and its motion vectors, from which can be retrieved several future frames [4]. Motion estimation algorithms focus on maximising the quality of the prediction while maintaining high speed in processing. In this context, motion estimation can be seen as a search problem [5].

In this paper, the performance of various block-matching algorithms, using quality of prediction and efficiency, is evaluated. Although new BMAs have been introduced, we have not found an extensive comparison among different methods. In [6]–[8] present comparisons based on the Peak Signal-to-Noise Ratio (PSNR) and the number of explored blocks (EXB). A similar comparison is presented in [9], along with measuring the complexity as the ratio of the time that takes the Full-Search (FS) to the time that takes the evaluated algorithm. In [10] the authors refer largely to hardware. They used the three criteria to compare various block-matching algorithms: silicon area, input/output requirement, and image quality.

The comparison is based on the following criteria: the PSNR, the EXB and the Structural Similarity Index (SSIM). Obtained results showed the Multi-Directional Gradient Descent Search (MDGDS) and the Fast Directional Gradient Descent Search (FDGDS) as the algorithms with the best quality prediction among all analysed algorithms. Also, the FDGDS has a balanced between high quality of prediction and low response time. The HEXBS has shown to be less sensitive to variations in the block size, than the other analysed

978-1-4673-1476-3/12/$31.00 ©2012 IEEE

algorithms.

The rest of the paper is organised in the following sections. Section II presents the algorithms under comparison. Section III contains the comparison criteria, including parameters and characteristics used in the implementation. Section IV is focused on the obtained results and Section V includes final comments as conclusions.

II. Selected Algorithms

Seven block-matching algorithms are selected. They cover different search strategies, for motion estimation in a video sequence. The selected algorithms are described as follows.

A. Full-Search (FS)

The Full-Search algorithm evaluates all positions in the window search of $(2W+1)\text{x}(2W+1)$[1] size. The smallest distortion calculated between the reference block and each block in the window search is used to determine the best matching.

The FS is by nature a brute force algorithm and involves a high computational cost. However, it is simple and guarantees a high accuracy in finding the best match [11].

B. Three-Step Search (TSS)

The TSS algorithm starts with evaluating the distortion in the central block and eight blocks around it, at an initial distance in pixels. The best candidate is taken as a new search center and eight block neighbours are selected around it, at half of the initial distance. This process is repeated until the distance is equal to one [12].

C. Four-Step Search (4SS)

The 4SS starts with evaluating the distortion in the central block and eight blocks around it, at an initial distance of two pixels. The best matched is calculated and eight new neighbours are selected around the best matched block – also at a distance of two pixels. Finally, the previous best matched block is used to explore the eight blocks around it at a distance of one pixel, and return the best of them [13].

D. Diamond Search (DS)

The DS follows the same strategy of the 4SS, but the selection of the eight neighbours is done in a diamond shape. Thus, in the final step, it only evaluates four blocks instead of eight [14].

E. Hexagonal Block Search (HEXBS)

The HEXBS is similar to the 4SS, but varies in the search pattern: starts with evaluating the distortion in the central block and six blocks around it (in hexagon shaped) at an initial distance of two pixels. In the last stage, it evaluates four neighbours using cross shaped pattern at a distance of a pixel. This algorithm has as an advantage that moving in either direction only explore three new blocks. Also, the HEXBS is faster than the DS, but has a lower quality of prediction in most cases [6].

[1]Let W be the block size.

F. Multi-Directional Gradient Descent Search (MDGDS)

The MDGDS algorithm starts with evaluating the distortion in the central block and independently in each of the eight surrounding directions: upper, lower, left, right, upper-left, upper-right, lower-left and lower-right directions. It makes a straight path, pixel by pixel in each of these directions if in each step the distortion is reduced, otherwise, the search stops in that direction. The algorithm stops when there is not reduction in the distortion in any of the eight directions mentioned above [7].

G. Fast Directional Gradient Descent Search (FDGDS)

The FDGDS is an improvement of the MDGDS that increases the speed of the algorithm and leads to little loss in quality of prediction.

The improvement is by detecting when – in a direction – a minimum is clearly better than the current search center. Thus, the algorithm stops to evaluate the remaining directions and starts a new stage in the minimum search found.

Po *et al.*, 2009 [8] proposes the measure Relative Ratio Distortion (RDR) as a criterion to determine whether a particular block is better than the reference block. When the path ends in one direction, the RDR is compared with a threshold and – if it is lower –, leaps and explores other directions.

III. Comparison Criteria

Since an algorithm A requires time proportional to the number of explored blocks (EXB), the efficiency of a BMA is determined by the EXB. The quality of prediction of a BMA is calculated using the Peak Signal-to-Noise (PSNR) and the Structural Similarity Index (SSIM).

The SAD is used as the Block Distortion Measure (BDM), which takes W^2 operations. The BDM is computed for each block on the analysed frame. In this way, the complexity of a BMA turns proportional to the number of explored blocks.

In the case of the PSNR, the signal will be the original frame and the noise will be the obtained reconstruction using the motion vectors. Let MAX_I be the maximum intensity, using 8-bits is 255, the PSNR is calculated as:

$$PSNR = 20\log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \tag{3}$$

Let x and y be two windows of common size $N\text{x}N$, the SSIM is calculated as:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{4}$$

where the mean intensity represents the luminance; the standard deviation represents the contrast; and C_1 and C_2 are included to avoid instability when $\mu_x^2 + \mu_y^2$ and $\sigma_x^2 + \sigma_y^2$ are very close to zero, respectively. The mean SSIM (MSSIM) is used to evaluate the overall image quality:

$$MSSIM(X,Y) = \frac{1}{M}\sum_{i=1}^{M} SSIM(x_i, y_i) \tag{5}$$

where X and Y are the reference and the distorted images, respectively; x_i and y_i are the image contents at the ith local window; and M is the number of local windows of the image [15].

The PSNR do not model the HVS (Human Visual System), the location of the errors is not taken into consideration, nor the sign of the error. In this way, the SSIM provides greater accuracy and consistency than the PSNR, which simply and objectively quantifies the error signal [16].

The experimental comparison was performed using the repository of freely redistributable test sequences, owned by Xiph.Org Foundation[2]. Table I shows the characteristics of the used sequences (see Fig. 1) in the study[3].

Three different block sizes are used in all algorithms: 8x8, 16x16 and 32x32. In this way, the FS algorithm was implemented with search windows of 17x17, 33x33 and 65x65 respectively.

TABLE I
CHARACTERISTICS OF THE TEST SEQUENCES

Video Sequence	Size	Number of Frames	Motion
Akiyo	352x288	300	Small
Coastguard	352x288	300	Large
Football	352x288	260	Large
Foreman	352x288	300	Medium
Garden	352x240	115	Medium
Mobile	352x288	300	Medium
Mother_daughter	352x288	300	Small
Silent	352x288	300	Small
Stefan	352x240	300	Large

The implementation was done using C++ programming language. Technical features of the PC – on which the tests were carried out – are: memory 3.4 GB; and four processor Intel(R) Core(TM)2 Quad CPU Q6600 @ 2.40 GHz.

IV. RESULTS

Tables II, III, and IV show obtained results of the comparison based on the reconstructing frames of selected video sequences, using differents block sizes and the full video sequences.

The HEXBS, the DS and the TSS are the algorithms with the lowest prediction quality.

When going through the results, some algorithms clearly stand out. The most efficient algorithms are the HEXBS, the MDGDS and the FDGDS, independent of the motion of the video sequence. The HEXBS produces low quality of prediction, while the two others have the best quality prediction among all analysed algorithms. Figs. 2, 3, and 4 show the SSIM performance of these algorithms in video sequences with small-, medium- and large-motion, respectively, using the first 50 frames and a block size of 16x16 pixels. In the three cases, the results generated by the FDGDS are ver close to those generated by the MDGDS. On the other hand, the low quality of prediction of the HEXBS is observed in the results of SSIM using Foreman and Stefan with medium- and large-motion.

[2]All video sequences used are in uncompressed format: YUV4MPEG, and are available at: http://media.xiph.org/video/derf/.

[3]The classification of the video sequences is taken from [5].

Fig. 1. Test sequences used in the study: (a) Akiyo, (b) Coastguard, (c) Football, (d) Foreman, (e) Garden, (f) Mobile, (g) Mother_daughter, (h) Silent, and (i) Stefan.

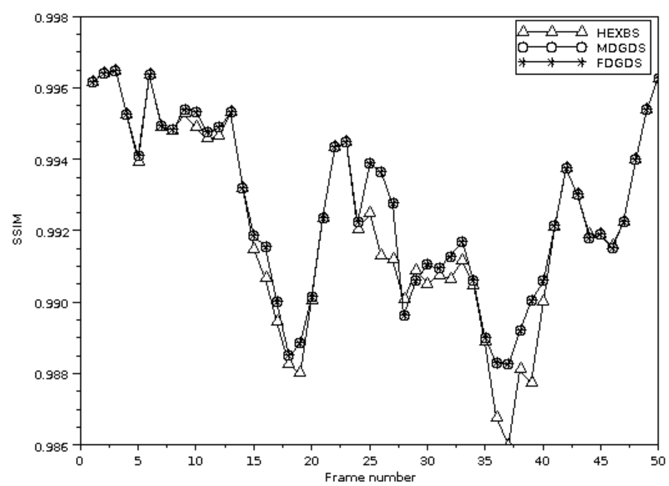

Fig. 2. SSIM performance of the HEXBS, the MDGDS, and the FDGDS on the first 50 frames of the Akiyo video sequence.

As the block size grows, the prediction quality tends to be lower (see Fig. 5). Increasing a block size covers larger areas of the frame, intensity values in the block will have larger variability. Using large block sizes decreases the quality of prediction obtained in video sequences with medium- and large-motion. However, in video sequences with small-motion, the use of large block sizes has low negative impact on the quality of prediction.

Some algorithms are more sensitive to the variation in

TABLE II
COMPARISON OF SELECTED ALGORITHMS USING BLOCK SIZE OF 8X8 PIXELS

Sequence	Measure	FS	TSS	4SS	DS	HEXBS	MDGDS	FDGDS
Akiyo	PSNR	43.615	43.116	43.290	42.893	42.991	43.583	43.566
	SSIM	0.996	0.993	0.993	0.992	0.992	0.993	0.993
	EXB	275.424	24.101	16.467	12.639	10.654	8.946	8.888
Coastguard	PSNR	31.616	30.359	31.362	28.922	31.103	31.358	31.336
	SSIM	0.926	0.905	0.922	0.865	0.916	0.923	0.923
	EXB	275.424	24.186	19.310	15.086	10.672	15.157	13.007
Football	PSNR	26.654	25.218	25.582	25.770	22.043	25.898	25.749
	SSIM	0.782	0.761	0.770	0.775	0.661	0.782	0.775
	EXB	275.424	24.225	24.802	19.453	10.677	27.345	23.671
Foreman	PSNR	32.961	31.512	32.043	31.375	30.062	32.518	32.344
	SSIM	0.936	0.894	0.906	0.891	0.864	0.913	0.910
	EXB	275.424	24.168	20.852	16.119	10.665	19.085	16.593
Garden	PSNR	25.233	22.092	24.143	20.792	23.312	24.690	24.615
	SSIM	0.922	0.833	0.886	0.791	0.860	0.913	0.910
	EXB	273.945	24.105	19.818	15.716	10.646	16.558	12.945
Mobile	PSNR	25.584	24.318	24.709	24.098	24.639	25.416	25.392
	SSIM	0.920	0.890	0.904	0.885	0.904	0.920	0.920
	EXB	275.424	24.129	17.458	13.493	10.660	12.231	11.986
Mother_daughter	PSNR	41.341	40.761	41.018	40.551	40.442	41.153	41.120
	SSIM	0.975	0.971	0.972	0.970	0.969	0.973	0.973
	EXB	275.424	24.158	17.900	13.988	10.667	18.527	18.259
Silent	PSNR	37.657	36.816	37.068	36.844	34.974	37.274	37.179
	SSIM	0.969	0.964	0.966	0.964	0.958	0.967	0.967
	EXB	275.424	24.101	17.472	13.431	10.655	11.319	10.729
Stefan	PSNR	24.734	23.441	23.149	23.263	21.667	23.059	23.034
	SSIM	0.875	0.795	0.794	0.798	0.723	0.795	0.793
	EXB	273.945	24.065	20.640	16.462	10.634	17.390	15.882

TABLE III
COMPARISON OF SELECTED ALGORITHMS USING BLOCK SIZE OF 16X16 PIXELS

Sequence	Measure	FS	TSS	4SS	DS	HEXBS	MDGDS	FDGDS
Akiyo	PSNR	42.944	42.821	42.858	42.541	42.586	42.940	42.939
	SSIM	0.993	0.992	0.992	0.992	0.992	0.993	0.993
	EXB	984.919	23.212	15.845	12.219	10.313	8.563	8.518
Coastguard	PSNR	30.477	30.265	30.347	28.670	30.341	30.371	30.369
	SSIM	0.914	0.908	0.911	0.866	0.904	0.911	0.911
	EXB	984.919	23.375	18.593	14.238	10.350	14.125	12.218
Football	PSNR	25.673	23.606	24.289	24.397	21.151	24.515	24.456
	SSIM	0.796	0.704	0.731	0.734	0.609	0.742	0.738
	EXB	984.919	23.460	25.736	20.087	10.357	28.560	25.769
Foreman	PSNR	32.119	30.717	31.345	30.594	29.305	31.392	31.305
	SSIM	0.923	0.886	0.899	0.885	0.849	0.905	0.902
	EXB	984.919	23.330	20.666	15.848	10.336	19.271	16.941
Garden	PSNR	23.794	22.980	23.326	21.276	22.653	23.579	23.547
	SSIM	0.903	0.862	0.877	0.812	0.849	0.894	0.892
	EXB	973.703	23.203	18.816	14.550	10.294	15.249	12.180
Mobile	PSNR	24.588	24.267	24.365	24.061	24.243	24.521	24.519
	SSIM	0.905	0.898	0.901	0.892	0.899	0.905	0.905
	EXB	984.919	23.228	16.296	12.479	10.318	11.231	11.065
Mother_daughter	PSNR	40.473	40.232	40.328	40.007	39.854	40.373	40.365
	SSIM	0.970	0.968	0.9689	0.970	0.966	0.969	0.969
	EXB	984.919	23.291	16.904	13.174	10.332	10.994	10.826
Silent	PSNR	35.973	35.241	35.437	35.308	33.942	35.421	35.395
	SSIM	0.961	0.958	0.959	0.957	0.953	0.960	0.960
	EXB	984.919	23.217	16.732	12.880	10.314	10.499	10.119
Stefan	PSNR	24.104	22.477	22.349	22.431	20.868	22.216	22.212
	SSIM	0.850	0.765	0.773	0.778	0.688	0.769	0.769
	EXB	973.703	23.152	20.464	16.434	10.275	16.671	15.438

TABLE IV
COMPARISON OF SELECTED ALGORITHMS USING BLOCK SIZE OF 32X32 PIXELS

Sequence	Measure	FS	TSS	4SS	DS	HEXBS	MDGDS	FDGDS
Akiyo	PSNR	42.177	42.177	42.177	42.099	42.097	42.177	42.177
	SSIM	0.992	0.992	0.992	0.991	0.991	0.992	0.992
	EXB	3425.970	21.485	14.682	11.430	9.646	7.943	7.916
Coastguard	PSNR	29.290	29.221	29.226	28.021	29.070	29.261	29.261
	SSIM	0.894	0.891	0.892	0.852	0.884	0.891	0.891
	EXB	3425.970	21.7999	17.257	13.106	9.723	12.837	11.206
Football	PSNR	23.683	22.044	22.724	22.793	20.478	22.793	22.781
	SSIM	0.7343	0.647	0.688	0.688	0.567	0.693	0.692
	EXB	3425.97	21.958	25.257	19.681	9.731	27.176	25.282
Foreman	PSNR	30.239	29.460	29.799	29.474	28.436	29.915	29.881
	SSIM	0.903	0.868	0.885	0.873	0.833	0.890	0.888
	EXB	3425.970	21.684	19.477	14.919	9.693	18.059	16.188
Garden	PSNR	16.853	16.798	16.818	16.379	16.693	16.833	16.832
	SSIM	0.811	0.795	0.802	0.752	0.778	0.808	0.807
	EXB	3231.180	21.145	16.937	12.840	9.352	13.094	10.655
Mobile	PSNR	23.593	23.579	23.585	23.513	23.532	23.590	23.590
	SSIM	0.884	0.883	0.884	0.881	0.882	0.884	0.884
	EXB	3425.970	21.495	14.921	11.450	9.651	10.086	9.968
Mother_daughter	PSNR	39.373	39.299	39.316	39.147	39.070	39.323	39.323
	SSIM	0.964	0.964	0.964	0.963	0.962	0.964	0.964
	EXB	3425.970	21.538	15.206	11.831	9.663	9.250	9.154
Silent	PSNR	33.781	33.431	33.502	33.453	32.811	33.423	33.423
	SSIM	0.950	0.949	0.950	0.949	0.946	0.951	0.951
	EXB	3425.970	21.495	15.389	11.948	9.648	9.286	9.148
Stefan	PSNR	15.505	14.947	15.020	15.073	14.557	14.987	14.987
	SSIM	0.787	0.676	0.694	0.702	0.602	0.687	0.6867
	EXB	3231.180	21.043	18.948	15.437	9.313	14.891	13.997

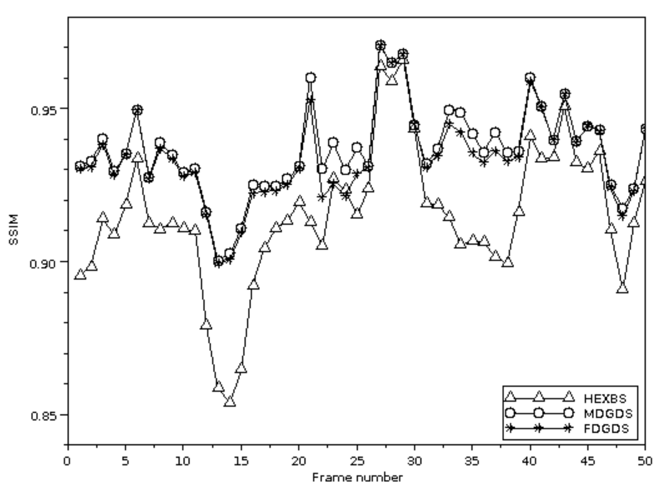

Fig. 3. SSIM performance of the HEXBS, the MDGDS, and the FDGDS on the first 50 frames of the Foreman video sequence.

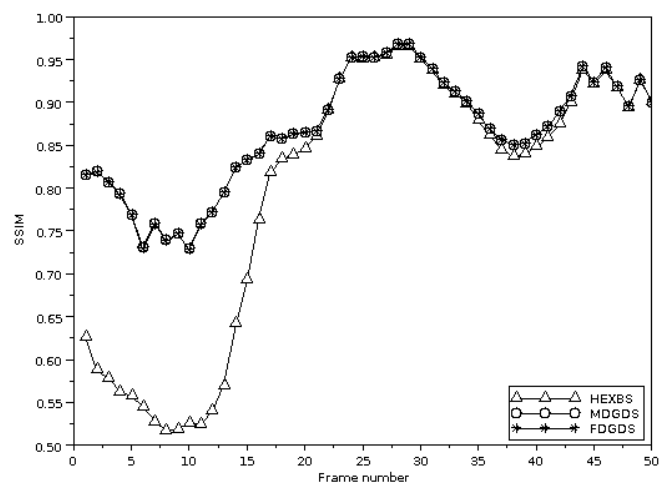

Fig. 4. SSIM performance of the HEXBS, the MDGDS, and the FDGDS on the first 50 frames of the Stefan video sequence.

block size than others. For all algorithms, the best quality of prediction was obtained using block size of 8x8 pixels. In Garden and Stefan video sequences the use of 32x32 pixels block shows that, regardless of the BMA used, results are quite distant from those obtained with block sizes of 8x8 and 16x16 pixels. The HEXBS is less affected by variation in block size. The quality of prediction obtained using block sizes of 16x16 and 32x32 pixels are very close to those obtained with block size of 8x8 pixels. In the other algorithms, the results obtained using 32x32 pixels block are close to those obtained using 8x8 pixels block in video sequences: Akiyo, Mobile and Mother_daughter. The TSS, the 4SS and the DS, the results obtained with block size of 16x16 pixels are close to those obtained with block size of 8x8 pixels in the video sequences: Akiyo, Coastguard, Foreman, Garden, Mobile, Mother_daughter, and Stefan. This is also true for the FDGDG and the MDGDS using video sequences: Akiyo, Coastguard, Mother_daughter, and Stefan.

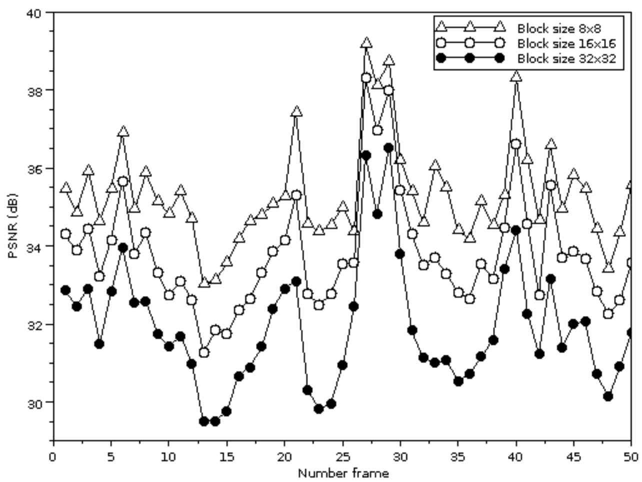

Fig. 5. PSNR performance of the MDGDS algorithm on the first 50 frames of the Foreman video sequence.

The Zero Motion Vector (ZMV) is the case when there is not movement between two consecutive frames of a video sequence. The number of explored blocks for the TSS are 25, for the 4SS 17, for the DS are 13, for the HEXBS 11, and for the MDGDS and the FDGDS are 9. The lasts algorithms only analyse the central block and eight blocks around.

V. Conclusions

In this paper, the performance of seven block-matching algorithms is assessed using video sequences with three different types of motion. This information may be used to determine which specific algorithm performs better depending on the motion of a sequence.

Among evaluated algorithms, the HEXBS is efficient but producd low quality of prediction, the MDGSD and the FDGDS are efficient and produce the best quality prediction. Moreover, the FDGDS generates close results to the MDGDS in terms of quality, but it explores less blocks than the MDGSD. This makes the FDGDS a balanced algorithm between high quality of prediction and low computation cost.

The HEXBS is less affected by the variation in the block size. While the other analysed algorithms show great loss of prediction by increasing the block size used.

References

[1] J. Watkinson, *MPEG Handbook*, 2nd ed. Focal Press, 2003.

[2] Y.-C. Lin and S.-C. Tai, "Fast full-search block-matching algorithm for motion-compensated video compression," *IEEE Trans. Commun.*, vol. 45, no. 5, pp. 527–531, 1997.

[3] B. Xiong and C. Zhu, "A new multiplication-free block matching criterion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 10, pp. 1441–1446, 2008.

[4] A. Gyaourova, C. Kamath, and S.-C. Cheung, "Block matching for object tracking," Lawrence Livermore National Lab., Tech. Rep., Oct. 2003.

[5] V. Padilla, "Algoritmos de block-matching para compresión de video," Final Career Project, Systems Engineering Program, Universidad del Valle, 2009.

[6] C.-H. Cheung and L.-M. Po, "Novel cross-diamond-hexagonal search algorithms for fast block motion estimation," *IEEE Trans. Multimedia*, vol. 5, no. 1, pp. 16–22, 2005.

[7] L.-M. Po, K.-H. Ng, K.-M. Wong, and K.-W. Cheung, "Multi-direction search algorithm for block-based motion estimation," in *IEEE Asia Pacific Conf. in Circuits and Systems (APPCAS)*, 2008, pp. 1466–1469.

[8] L.-M. Po, K.-H. Ng, K.-W. Cheung, K.-M. Wong, Y. Uddin, and C.-W. Ting, "Novel directional gradient descent searches for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 8, pp. 1189–1195, 2009.

[9] M. Ghoneim, N. Tsumura, T. Nakaguchi, T. Yahagi, and Y. Miyake, "A fast block matching algorithm based on motion vector correlation and integral projections," *IEICE Transactions*, vol. 92-D, no. 2, pp. 310–318, Feb. 2009.

[10] S.-C. Cheng and H.-M. Hang, "A comparison of block-matching algorithms mapped to systolic-array implementation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 5, pp. 741–757, 1997.

[11] J. Huska and P. Kulla, "Trends in block-matching motion estimation algorithms," Dept. of Radioelectronics, Slovak Univ. of Technology, Bratislava, Tech. Rep.

[12] L.-M. Po, K.-H. Ng, K.-M. Wong, and K.-W. Cheung, "Motion compensated interframe coding for video conferencing," in *Proc. Nat. Telcommun. Conf.*, Nov. 1981, pp. G5.3.1–G5.3.3.

[13] L.-M. Po and W.-C. Ma, "A novel four-step search algorithm for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 3, pp. 313–317, 1996.

[14] J. Y. Tham, S. Ranganath, M. Ranganath, and A. Kassim, "A novel unrestricted center-biased diamond search algorithm for block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 4, pp. 369–377, 1998.

[15] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004.

[16] C. S. varnan, A. Jagan, J. Kaur, D. Jyoti, and D. S. Rao, "Image quality assessment techniques pn spatial domain," *International Journal on Computer Science and Technology*, vol. 2, no. 3, pp. 177–184, Sep. 2011.