QUASI-RANDOM SEARCH ALGORITHM FOR FAST MOTION ESTIMATION

Robson Lins[*], *Diogo B. Henriques*[*], *Emerson Lima*[†] *and Silvio Melo*[*]

[*]Federal University of Pernambuco, Computer Center, University City - 50740-560 - Recife / PE, Brazil
[†]University of Pernambuco, Agamenon Magalhães Avenue - 50100-010 - Recife / PE, Brazil

ABSTRACT

In this paper, we propose a new Motion Estimation algorithm based on low discrepancy sequences, essentially quasi-random sequences used as point sampling to compute multivariate integral approximations by methods such as Monte-Carlo. Our evaluation of the proposed method took into consideration PSNR mean values, computational effort and bit-rate. The results support that the proposed algorithm entitled Quasi-Random Search (QRS) is able to significantly reduce computational effort by 73,06% in average when compared to the UMHS algorithm while maintaining video quality and a slight increase in the bit-rate.

Index Terms— motion estimation, video coding, block-matching, low discrepancy sequences

1. INTRODUCTION

Motion Estimation (ME) is a simple, elegant and efficient manner to identify and express motion and, therefore, is frequently chosen by video coding standards such as MPEG-4, H.264/AVC and the emerging H.265/HEVC [1, 2].

ME methods identify blocks in the current frame that best match blocks on the previous frame (reference frame). In general terms, ME uses a search window specified in the reference frame for each block within the current frame. Since there is a greater likelihood that similar blocks are near the block position, the search window must contain a region centered in the blocks original position. Each block within the search window is named a candidate block for any of these blocks may be chosen as the best match by the ME.

Consider a block $M \times N$ and a search window $(2dm + 1) \times (2dm + 1)$, where dm is the maximum displacement. The current frame's blocks are compared with the reference frame's blocks and compared using some similarity measure. One such measure for example is the *Sum of Absolute Differences* (SAD) expressed by:

$$\mathrm{SAD}(i, j) = \sum_{m=1}^{M} \sum_{n=1}^{N} |s_k(m, n) - s_{k-1}(m+i, n+j)|, \quad (1)$$

where $s_k(m, n)$ is the intensity of the current block in the position (m, n) and $s_{k-1}(m + i, n + j)$ is the intensity of the candidate block in the position $(m + i, n + j)$, where i and j are the candidate block's displacement in the search window. The candidate block with the least SAD is chosen as the best match, specifying the motion vector. According to [3], nearly 89,2% of the computational effort of the coding process relates to the ME. Therefore, reducing the ME computational effort is a topic which inspires great interest.

In order to reduce ME's computational effort, fast search algorithms have been developed [4, 5, 6, 7, 8]. These algorithms reduce the number of potential candidate blocks while aiming for a high coding efficiency. The *Unsymmetrical-cross Multi-Hexagon grid Search* (UMHS) algorithm that effectively reduces the number of candidate blocks within the search window has been used by H.264/AVC Reference Software [5].

In a related work, [9] presents an statistical analysis on the performance of the UMHS algorithm and introduces an early termination step based on adaptive threshold. These thresholds both maintain the rate-distortion (RD) [10] optimization while reducing significantly the ME's processing time. As a result, the method proposed by [9] was adopted by H.264/AVC Reference Software. The results presented in [9] are compared to the proposed method in this paper (see Section 4). This work focuses in the reduction of computations effort by ME with the usage of low discrepancy sequences which presents a uniform coverage throughout the region as well as a faster computation of the similarity measure used by block matching.

Another work focused on computation effort reduction for ME based on adaptive threshold early termination was presented by [11]. In addition to the early termination, [11] proposed a division of the search window into sectors according to the occurrence probability in a way that allowed the algorithm to satisfy the termination criteria, a low threshold, quicker. The sectors' occurrence probabilities were computed by the H.264/AVC Reference Software based on the statistical characteristics of RD cost. This evaluation was also used to determine the threshold value.

The work presented in [12] proposed a new ME algorithm using a non-translational movement type. The movement model chosen is based on image recording techniques widely used in computer vision and augmented reality [13, 14]. Not only did the authors use the H.264 block partitioning system, but also 32×32, 32×16 and 16×32 block sizes. As for RD cost, the algorithm establishes a decision process between translational and non-translational movement types. In this particular case, the lowest RD cost is chosen, this implies in a greater complexity cost for the motion vector computation.

A ME algorithm entitled *Simulated Annealing Adaptive Search* (SAAS) proposed by [8] is based on the statistical analysis of the motion vectors coupled with the *Simulated Annealing* concept. The SAAS algorithm divides the search window into twenty-four regions where each region defines a search direction. The discussed work uses a initial prediction similar to the UMHS algorithm as well as its local refinement.

The work in [15] presents a new ME method based on adaptive maximum displacement. If the maximum displacement value is not correctly specified, some ME methods are unable to properly encode the input sequence, presenting image quality loss. To overcome this difficulty, the authors propose to consider vertical and horizontal movement separately resulting in a rectangular search window. The method was evaluated using the H.264/AVC Reference Software and the results shows a insignificant loss to PSNR.

Taking into consideration the referenced works, one can perceive that multiple strategies are common among them. In this work, a new search pattern for ME is proposed seeking to reduce the computational effort while maintaining visual quality. In order to obtain optimum results, the similarity measure has been enhanced to work with a pre-computed lattice subsampling based on low discrepancy sequences. All the proposed solutions presented in this work are compatible with the techniques discussed in this section. Such solutions make use of low discrepancy sequences, quasi-random sequences generally used to reduced the number of sampled points to compute approximate values of multiple integrals by methods like Monte-Carlo [16].

2. LOW DISCREPANCY SEQUENCES

Monte Carlo methods form a class of algorithms that use pseudo-random numbers in tasks that usually involve numerical processing of a large amount of data [16]. These methods have been applied to a great variety of numerical problems such as in the iterated integral estimate, and usually present great advantages over the traditional approaches of partition intervals such as Simpson's and the like [17].

Among the pseudo-random sequences, the so called low discrepancy sequences have been the object of major research, given its apparent superiority with respect to space uniform cover and the convergence performance in the integral estimates. We call *discrepancy* the measure of uniformity of a given sequence[1]. More formally:

Definition 1 (Discrepancy) *Let* $\omega = \{x_1, x_2, \cdots, x_n, x_{n+1}, \cdots\}$ *an infinite sequence of real numbers in the interval* $[0,1]$ *and let* $I \subseteq [0,1]$ *one sub-interval. We define* $A(I;n)$ *the amount of points of the subsequence* $x_1, x_2, \cdots, x_n$ *belonging to I, i.e,* $A(I;n) = |I \cap \{x_1, x_2, \cdots, x_n\}|$*. In a uniform sequence, the amount of points* $A(I;n)$ *is proportional to I's measure, i.e,* $\frac{A([\alpha,\beta);n)}{n} = \beta - \alpha$ *considering that* $I = [\alpha, \beta)$*. The measure of the deviation*

$$D_n = D_n(\omega) = \sup_{0 \leq \alpha < \beta \leq 1} \left| \frac{A([\alpha,\beta);n)}{n} - (\beta - \alpha) \right|$$

is called Discrepancy *of the first* n *points of the sequence* ω*.*

The computation of D can be quite complicate. An associate measure of a somewhat simpler computation is the so called *star-discrepancy* expressed by:

$$D_n^* = D_n^*(\omega) = \sup_{0 < \beta \leq 1} \left| \frac{A([0,\beta);n)}{n} - \beta \right|.$$

A well established result relates both measures through the inequality

$$D_n^* \leq D_n \leq 2D_n^*.$$

Considering sequences in the interval $[0,1]$, if the discrepancy of the sequence is zero, then the cover of the sequence is completely uniform and if it is near 1 then the sequence will be poorly uniform, leaving empty chunks in the interval. Some sequences may be regarded as a low discrepancy, i.e., near zero discrepancy, when considered as a whole, however they may present high discrepancy behavior during their early stages of construction. For instance, consider a total of 256 points for a given low discrepancy sequence, but it is possible that, if we take an intermediate phase of its construction, encompassing 64 points for example, this sequence may present a poorly uniform cover, which is a high discrepancy behavior. Other sequences possess the property of behaving like a low discrepancy sequence in all of its construction phases, such as Van der Corput-Halton's. In this paper, we will utilize Van der Corput-Halton's sequences [16], defined as follows:

Definition 2 (Van der Corput-Halton's Sequence (VDH)) *Let* $b > 1$ *a positive integer number. The sequence* $VDH_b = \{x_1, x_2, \cdots\}$ *for which the term* x_i *is defined by*

$$x_i = \sum_{j=0}^{s} \frac{a_j}{b^{j+1}},$$

where $\sum_{j=0}^{s} a_j b^j$ *is the expansion in the numerical base* b *of the number* $i-1$ *is called* Van der Corput-Halton's Sequence in the base b.

2.1. The VDH sequence and block-matching

Let X and Y blocks of size $M \times N$ in the current frame and the search window in the previous frame, respectively. Let $P = \{p_1, p_2, \cdots, p_k\}$ be the subset of $\{1, 2, \cdots, M\} \times \{1, 2, \cdots, N\}$, i.e, a subset of all possible coordinates of blocks X e Y. We define the mean absolute difference *induced* by P as:

$$\mathrm{SAD}_P(X,Y) = \sum_{l=1}^{k} |X(p_l) - Y(p_l)|. \qquad (2)$$

---

[1] The definitions and notation that follow are the same as in [16].

Notice that if $P = \{1, 2, \cdots, M\} \times \{1, 2, \cdots, N\}$ then Equation (2) corresponds to Equation (1) for fixed positions (m, n) and (i, j). Because summations and integration are related operations, the use of Monte Carlo techniques for the summations' evaluation, just like it is done with integration, is suggested in this work. In particular, the use of low discrepancy sequences in the fast evaluation of the summations involved in the computation of the SAD, as will be seen briefly, presented very encouraging results.

In order to employ low discrepancy sequences making use of the concept of induced SAD, we need to transform the sequence VDH $_b$ of real numbers in the interval $[0, 1)$ into a sequence P of integers lying in the set[2] $\{1, 2, \cdots, m \times n\}$.

One approach consists of taking P as the ordering of $\{1, 2, \cdots, m \times n\}$ induced by the indexing of VDH$_b$, i.e., $p_i = j$ if and only if x_i is the j-th element in the ordered list of VDH$_b$'s elements. For instance, the sequence $A = \{0.8, 0.2, 0.7, 0.4\}$ induces, through this mechanism, the sequence $B = \{4, 1, 3, 2\}$, meaning that in the ordered listing of A its original elements hold respectively the fourth, first, third and second positions. Through this approach the repetitions are avoided but the convergence is preserved.

Fig. 1 shows the first eight points in a 2D VDH sequence in bases 2 and 3 for a 8×8 pixel block. From the image processing point of view, this represents a pixel subsampling, technique used to accelerate the computation of similarity measures such as the SAD within ME. The SAD computation takes into consideration only the painted dots, while the white dots are ignored. Figure 1 presents a VDH-based subsampling with an 8:1 ratio where only one eight of the pixels are used by the similarity measure.

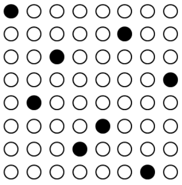

Fig. 1. 8:1 VDH subsampling.

In order to preserve the hierarchical structure used by the H.264/AVC coding standard, we used a latticed pattern subsampling as shown in Fig 2. Therefore, block partitions of sizes 16×16, 16×8, 8×16 and 8×8 can be used to generate the motion vectors. Thus, the low discrepancy sequence can be computed only once and used as a mapping for a subsampling method to reduce computational effort related to the similarity measure even in cases of multiple block types such as the H.264/AVC and H.265/HEVC.

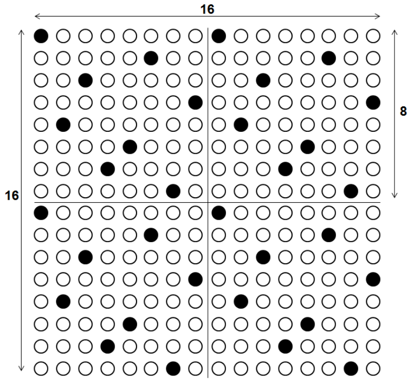

Fig. 2. Hierarchical VDH subsampling lattice.

3. QUASI-RANDOM SEARCH (QRS)

As seen in Section 2.1, consider k points in a 2D VDH sequence on bases 2 and 3, in other words the sequence $\{P_l\}_{l=1 \cdots k} = \{(a_l, b_l)\}_{l=1 \cdots k}$ where $a_1, a_2, \cdots, a_k$ is the base 2 sequence and $b_1, b_2, \cdots, b_k$ the base 3. The algorithm entitled *Quasi-Random Search* (QRS) proposed in this work uses a VDH sequence to select the candidate blocks within the search window. In order to select such blocks, a normalization of $a_1, a_2, \cdots, a_k$ to the $[1, 2dm + 1]$ interval and $b_1, b_2, \cdots, b_k$ to the $[1, 2dm+1]$ is required. In this work, we used a sixty-four point sequence beginning from the seventh point in a spiral around the search window's origin point as seen as blue squares in Figure 3.

In order to efficiently generate the motion vectors in video sequences containing slow movements, the QRS algorithm uses a NTSS-inspired search in its first step [4] in a 5×5 squared area centered in the origin (squared block in the center of Fig. 3). We then apply an early termination check as UMHS' algorithm. If this condition is satisfied, the algorithm skips to the last step, where a local refinement is used composed of a small hexagon search followed by a small diamond pattern (see Figure 3). The position with a least SAD$_P$ value in the local refinement, will be used to generate the motion vector. If the early termination criteria is not met, in the second step, another sixty-four candidate blocks are evaluated in a four-group-of-sixteen schema. For every group of sixteen blocks that is evaluated, an early termination check is performed and if met, the algorithm jumps to the last step and the position with the smallest SAD$_P$ is chosen to generate the motion vector.

Fig. 4 shows the block diagram for the QRS algorithm. Note that the initial prediction as well as the early termination check is used to reduce the actual number of candidate blocks checked.

---

[2] in fact, the sequence P in the definition of induced metric is a subset of $\{1, 2, \cdots, m\} \times \{1, 2, \cdots, n\}$. By using the standard linearization of a matrix, it is enough to consider P as a subset of $\{1, 2, \cdots, m \times n\}$

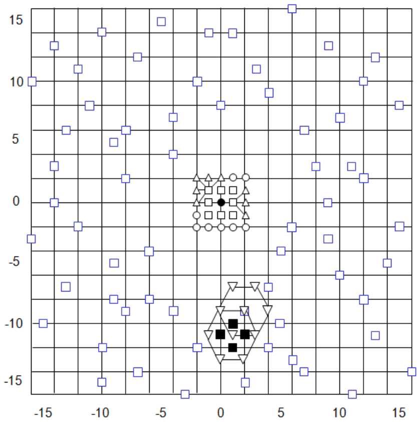

Fig. 3. QRS algorithm search pattern. In the 5×5 NTSS-inspired search the initial squared pixels centered around the black pixel is tested first. If the center pixel is not the optimum a second sub-step tests the immediate neighborhood shown by the triangles for both edge and corner cases. The local refinement slides the hexagonal shape and then a small diamond shape until the best match is the center pixel.

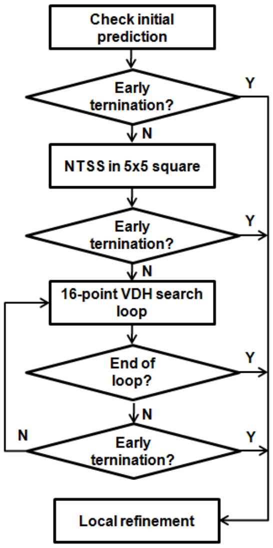

Fig. 4. QRS algorithm block diagram.

4. RESULTS

The VDH-based subsampling was encoded in the H.264/AVC Reference Software. This way, both the similarity measure SAD_P and the SAD could be used. In order to evaluate the performance of the motion estimation (ME) algorithms, we included a counter that increments every time a pixel comparison is made by the SAD. This allows the actual number of operations to be computed, not the theoretical or maximum one.

The codec was configured as such: 150 frames, one I-type frame for every four P-type frames (*IntraPeriod* = 4), RD optimization on, quantization parameter set to 28 (*QP* = 28) and four reference frames (*NumberReferenceFrames=4*). The maximum search range is configured according to the input sequence format, 32 for HD videos and 16 for CIF/QCIF. The PSNR was measured by the reference software. To evaluate the proposed method, we calculated the bit rate variation and the number of pixel comparisons made by the similarity measure (SAD) as follows:

$$\Delta Rate = \frac{Rate_{method} - Rate_{reference}}{Rate_{reference}} \times 100\% \text{ and } \quad (3)$$

$$\Delta OP = \frac{OP_{method} - OP_{reference}}{OP_{reference}} \times 100\%, \quad (4)$$

where $Rate_{method}$ and OP_{method} are the bit rate variation and the number of pixel comparisons of the proposed method (QRS). $Rate_{reference}$ and $OP_{reference}$ indicate the bit rate and number of pixel comparisons made by the UMHS method implemented in the H.264/AVC Reference Software.

Table 1 shows the results regarding image quality (PSNR), computational effort (OP) and coding bit-rate (Rate). Considering the PSNR, the QRS algorithm displays an insignificant quality loss when compared to the original UMHS, in the other hand, the computational effort has been greatly reduced by 73,06% in average and low fluctuations in the bit-rate.

Table 1. Experimental Results in the H.264/AVC for the QRS algorithm.

Sequence	ΔPSNR (dB)	ΔOP (%)	ΔRate (%)
Akiyo	-0,008	-76,60	0,23
Container	-0,003	-77,56	0,95
Coastguard	-0,01	-79,65	0,33
Foreman	-0,019	-74,97	1,28
Football	-0,026	-85,13	2,59
Mobile	0,004	-66,66	2,92
Soccer	-0,031	-82,97	4,38
Soccer (HD)	-0,022	-40,90	-1,51

5. CONCLUSION

The investigation on UMHS algorithm optimization with emphasis on low computational costs resulted in the development of a new ME algorithm, *Quasi-Random Search* (QRS). This algorithm presents a search pattern based on VDH low discrepancy sequences coupled with a pixel subsampling also based on VDH sequences. In addition, this algorithm uses early termination checks to accelerate the search procedure, as well as an initial prediction to avoid local minima traps and reduce the total number of iterations. In this way, quality losses are kept to an insignificant level while greatly increasing the algorithms performance.

6. REFERENCES

[1] Iain E.G. Richardson, "H.264 and MPEG-4 video compression. video coding for next-generation multimedia," in *Wiley*, 2005.

[2] G. Sullivan and J. R. Ohm, "Meeting report of the 10th meeting of the joint collaborative team on video coding," in *Joint Collaborative Team on Video Coding (JCT-VC)*, Stockholm, July 2012.

[3] Y. W. Huang, B. Y. Hsieh, S. Y. Chien, S. Y. Ma, and L. G. Chen, "Analysis and complexity reduction of multiple reference frames motion estimation in H.264/AVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 4, pp. 507–522, 2006.

[4] R. Li, B. Zeng, and L. M. Liou, "A new three-step search algorithm for block motion estimation," *IEEE Transactions on Circuits Systems for Video Technology*, vol. 4, no. 4, pp. 438–442, 1994.

[5] Zhibo Chen, Jianfeng Xu, Yun He, and Junli Zheng, "Fast integer-pel and fractional-pel motion estimation for H.264/AVC," *Journal of Visual Communication & Image Representation*, vol. 17, no. 2, pp. 264–290, 2006.

[6] C. Cai, H. Zeng, and S. K. Mitra, "Fast motion estimation for H.264," *Signal Processing: Image Communication*, vol. 24, no. 8, pp. 630–636, 2009.

[7] H. S. Lee, J. H. Jung, and D. J. Park, "2:1 candidate position subsampling technique for fast optimal motion estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 7, pp. 1052–1056, 2010.

[8] Z. Shi, W. A. C. Fernando, and A. Kondoz, "Adaptive direction search algorithms based on motion correlation for block motion estimation," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 3, pp. 1354–1361, 2011.

[9] X. Xu and Y. He, "Improvements on fast motion estimation strategy for H.264/AVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 3, pp. 285–293, 2008.

[10] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 74–90, 1998.

[11] M. G. Sarwer and Q. M. J. Wu, "Adaptive variable block-size early motion estimation termination algorithm for H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 8, pp. 1196–1201, 2009.

[12] A. A. Muhit, M. R. Pickering, M. R. Frater, and J. F. Arnold, "Video coding using elastic motion model and larger blocks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 5, pp. 661–672, 2010.

[13] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon, *Vision Algorithms: Theory and Practice*, Springer, 2000.

[14] S. Benhimane and E. Malis, "Real-time image-based tracking of planes using efficient second-order minimization," *Proc. International Conference on Intelligent Robots and Systems*, pp. 943–948, 2004.

[15] H. S. Kang, S. W. Lee, and H. G. Hosseini, "Probability constrained search range determination for fast motion estimation," *ETRI Journal*, vol. 34, no. 3, pp. 369–378, 2012.

[16] L. Kuipers and H. Neiderreiter, *Uniform Distribution of Sequences*, Addison Wesley Publishing Company, Inc., New York, 1992.

[17] F. James, "Monte carlo theory and practice," *Rep. Prog. Phys.*, vol. 43, pp. 1147–1188, 1980.