Sobol Partial Distortion Algorithm for Fast Full Search in Block Motion Estimation

Davide Quaglia and Bartolomeo Montrucchio

Dipartimento di Automatica e Informatica, Politecnico di Torino,
corso Duca degli Abruzzi 24,
I-10129 Torino (Italy)
`{davide.quaglia,montru}@polito.it`
`http://multimedia.polito.it/quaglia, http://www.polito.it/~montru`

Abstract. Block motion estimation is a cpu intensive task for video encoding. Many fast algorithms have been developed to solve this problem, trying to improve both block searching and block matching. Some of them reduce the image quality compared to the full search method in order to improve performance. The algorithm presented in this paper, called Sobol Partial Distortion (SPD) algorithm, is a full search (lossless), fast matching, block motion estimation algorithm, applying partial distortion elimination. It uses a new matching strategy to quickly compute distortion and reduce block matching computation. Image-dependent computation is not required, since the matching strategy does not depend on the frame sequence. The proposed algorithm performs well in terms of computational speedup in comparison with other existing full search algorithms.

1 Introduction

Motion compensation is used in video encoding to improve the efficiency of the prediction from past or future frames. Motion estimation [1][2] is the process of evaluating movements between adjacent frames. The so-called block matching algorithms are the most important of these estimation methods, especially in coding schemes based on the discrete cosine transform. Pel-recursive, frequency domain, and gradient motion estimation methods are less frequently used.

For each test block in the current frame, the block matching methods try to find the most similar candidate block in the previous frame. The displacement between these two blocks is the motion vector for the given test block. Unlike in pel-recursive methods, the motion vector refers to all pixels of the block.

The most accurate block matching method is the Full Search (FS), that compares every possible candidate block in the search window with the test block; in this way, it produces accurate results, but it is slow.

The block matching technique is used in well-known video encoding standards, such as MPEG-1, MPEG-2, MPEG-4, H.261 and H.263. In these standards, full search motion estimation requires up to 70% of the encoding time, and hardware and software encoders suffer from these modest speed performances.

For this reason, several alternative and faster techniques have been developed, which can be classified into two categories. The first is based on lossy motion estimation algorithms, with some degradation of the predicted image compared with FS; recent examples are [3–9]. The second is based on several lossless algorithms [10–15]. Particularly important among them are the Successive Elimination Algorithm (SEA) [10], an improved version of it called Extended SEA (ESEA) [16], and the Partial matching Distortion Elimination (PDE) [13], which also uses an adaptive matching scan and the selection of representative pixels [15].

Viewing the problem from another angle, two approaches can be chosen to reduce computation in block matching algorithms. The first reduces the number of candidate blocks in the search window (e.g. [3]), while the second reduces the number of pixels involved in each candidate-test block comparison (e.g. [6]). Both these methods in general are lossy; however lossless methods use the same concepts. SEA reduces safely the search space through the elimination of impossible candidates, and ESEA introduces tighter bounds for the sum of absolute differences (SAD) and exploits the already calculated lower bounds during the calculation of the matching criterion (ESEA also extends SEA for the application of the MSE). PDE allows stopping the candidate-test block comparison if the partial sum of matching distortion (SAD) is larger than the matching error for the most similar candidate block found so far. Picking candidates in a particular order within the search window often leads to finding the global minimum of distortion faster, so that many comparisons can be prematurely stopped. Therefore PDE has been improved both by changing the searching method with the SpiralPDE [13, 14], in which the searching order follows a center-biased spiral, and by changing the matching method [15]. In particular in [15] a strategy for finding representative pixels (and doing an adaptive matching scan) has been shown to further reduce the comparison time in block matching (with respect to SpiralPDE).
It is important to note that SEA and PDE can be combined together [7].

In [15] the order in which pixels are considered during block matching is a function of the frame content (through gradient). In this paper we will show that further improvements can be achieved using Sobol's sequence to order pixel checks within a block without needing to perform computations based on the image. This sequence is independent of the image and it can be precomputed to avoid any overhead.

Therefore, the algorithm presented in this paper is a new full-search (lossless), fast-matching, block motion estimation algorithm, using partial distortion elimination, called the Sobol Partial Distortion (SPD) algorithm.

Experimental results show that this way of comparing pixels during block matching improves computational performances up to 20% with respect to SpiralPDE.

Since the algorithm does not require any pre-processing step, its migration towards a hardware implementation could be easier, and will be the subject of future work.

This paper is organized as follows. Section 2 describes the basic idea used in the algorithm, while Section 3 introduces Sobol's sequence. Section 4 explains the SPD algorithm in detail, and the results are presented in Section 5. Finally, conclusions and future work are reported in Section 6.

2 The Basic Idea

The block matching operation is based on the assumption that all pixels in the block move by the same amount. Therefore a good motion estimation can be often obtained by using only a subset of representative pixels in the block. A possibly good way to identify representative pixels without additional computation is to use a pseudo-random uniform distribution. In particular, Sobol's distribution can be fruitfully employed since a block comparison can be stopped at any moment (applying the partial distortion elimination approach) without losing the property of uniformity. On the other hand, regular grids are more uniform than Sobol's sequence though they can not be stopped at any point without losing their properties.

In this paper we will show that Sobol's sequence in effect provides interesting results without any a priori knowledge of the image.

3 Sobol's Sequence

Sobol's pseudo-random sequence was first introduced for Monte Carlo integration by I. M. Sobol [17] in 1967. The sequence generates numbers between zero and one in an S-dimensional space; successive points uniformly fill a region of the space [18].

The sequence, scaled into a 16×16 block, is used to define the order in which pixels are checked within the block: this order is specified in Fig. 1.

The computation of the sequence is fast, although in this work the sequence is precomputed to avoid overheads. The main feature of Sobol's sequence is a more regular uniformity of points compared to a uniformly distributed random sequence. Only a regular grid has better uniformity, but a regular grid can not be prematurely stopped without altering its properties.

4 SPD Algorithm

The SPD algorithm is based on partial distortion elimination. According to this approach, the SAD evaluation for a block stops if the partial sum becomes larger than the matching error for the most similar candidate block found so far. The number of operations of this approach depends both on the visiting order of the pixels within a block to compute SAD (matching strategy) and on the comparison order of candidate blocks to the test block (search strategy). In fact, the more different are the pixels, the earlier the SAD computation will

133	129	125	251	167	41	213	83	102	228	24	154	206	76	176	50
237	107	143	17	69	195	55	185	136	10	246	116	32	162	94	220
179	53	201	71	27	157	97	223	210	80	172	46	122	248	4	134
89	215	35	165	241	111	139	13	60	190	66	192	148	22	234	104
203	77	177	47	99	229	25	151	170	40	212	86	2	128	124	254
33	159	91	221	137	7	243	117	68	198	58	184	236	110	146	16
119	249	5	131	207	81	173	43	30	156	96	226	182	52	200	74
149	19	231	105	61	187	63	193	240	114	142	12	88	218	38	164
101	227	23	153	205	75	175	49	0	130	126	252	168	42	214	84
135	9	245	115	31	161	93	219	238	108	144	18	70	196	56	186
209	79	171	45	121	247	3	255	180	54	202	72	28	158	98	224
59	189	65	191	147	21	233	103	90	216	36	166	242	112	140	14
169	39	211	85	1	127	123	253	204	78	178	48	100	230	26	152
67	197	57	183	235	109	145	15	34	160	92	222	138	8	244	118
29	155	95	225	181	51	199	73	120	250	6	132	208	82	174	44
239	113	141	11	87	217	37	163	150	20	232	106	62	188	64	194

Fig. 1. Sobol's sequence for a 16 × 16 grid

be stopped. Moreover, if the best candidate block is found early, then many comparisons will be skipped.

PDE matching strategy consists in the computation of the partial distortion row by row (where a row is made up of 16 pixels); the implementation of the search strategy is also row by row, from top to bottom. SpiralPDE (PDE improvement) search strategy follows a center-biased spiral, in order to exploit the center-biased distribution of motion vectors.

SPD algorithm involves:

- spiral search strategy in the search window,
- Sobol's sequence order (see Fig. 1) for the matching strategy (partial distortion computation),
- comparisons made for each group of 8 pixels (equivalent to a half row of SpiralPDE).

In PDE, the test on the partial distortion comparison is performed for each row of 16 pixels. Among possible choices (1,2,...,16 pixels), we have chosen 8-pixel-groups for Sobol's sequence in the SPD algorithm since this approach leads to the highest speedup over SpiralPDE as explained in Section 5 and shown in Fig. 3.

It is worth noting that our algorithm (like others derived from PDE) can be successfully combined with SEA or its improved versions [11] [16] to achieve maximum speedup.

5 Experimental Results and Remarks

We tested the performances of the SPD algorithm with respect to SpiralPDE. A large set of sequences has been encoded using a modified MPEG-2 encoder derived from ISO/IEC [13]. From the implementation point of view, in order to be fair, we have used the same programming techniques both for SpiralPDE and SPD; for this reason a comparison of the two algorithms is also feasible in terms of the total encoding time.

We have used a total of 17 sequences in various formats: QCIF (176×144), CIF (352×288) and CCIR-601 (720×486). They are well-known sequences, covering a large spectrum of applications from cinema to videoconference; some of them have large motion and others, commonly called "head-and-shoulders", are almost inactive. Sequences used are: "bream0", "bream1", "carphone", "claire", "container", "foreman", "garden", "glasgow", "grandmother", "miss america", "mobile", "mother&daughter", "news", "salesman", "silent", "suzie" and "trevor". The number of frames for each sequence varies from 80 up to 800. The search range is ± 15 pixels, while the block size is 16×16. We have used several Intel-based processors to test and evaluate our algorithm; in all cases results show the same behavior.

Results are reported in Figs. 2, 3 and Table 1.

Figure 2 shows averages and standard deviations for all sequences of the percentage reduction in numbers of rows checked before discarding the block. For SPD a row-equivalent value is obtained by dividing the number of checked pixels by 16. The gain is a maximum when the granularity of partial distortion is one pixel.

In Fig. 3 a similar plot is shown for the percentage of cpu time reduction for SPD with respect to SpiralPDE. Results are in percentage total cpu time for the encoding of all frames and are plotting for partial distortion comparison every 1,2,4,8,16 pixels. In average the advantage is a maximum for 8 pixels, although it depends slightly on the chosen video sequence. This maximum is due to the trade-off between the overhead produced by the number of comparisons (increasing toward left on the Figure) and by the number of pixels considered before the subsequent comparison (increasing toward right on the Figure).

Table 1 reports the average value of checked rows for SEA, PDE, SpiralPDE, the "Proposed 2" (P2) algorithm from [15] and SPD. In this case SPD does comparisons every 8 pixels and, therefore, a row-equivalent value is reported. Data for all algorithms are for these five sequences: "carphone", "claire", "foreman", "grandmother" and "trevor", common between [15] and our tests. Data for SEA, PDE, SpiralPDE and P2 are taken from [15]. SPD shows an average computational gain of up to 20% with respect to SpiralPDE algorithm.

It is worth remarking that results in Fig. 2 and Table 1 are different from those in Fig. 3: this is because the number of checked rows affects the motion estimation time which is only a part of the total encoding time considered in Fig. 3. Moreover Fig. 2 does not take into account the different comparison overhead with respect to SpiralPDE that does comparisons every 16 pixels.

We can conclude that the SPD algorithm performs well with respect to other algorithms; the method rejects impossible candidates faster than other PDE based algorithms, without any additional frame-based computation.

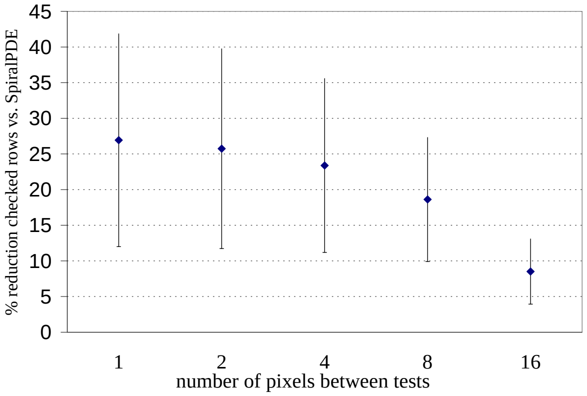

Fig. 2. Average value and standard deviation of the percentage of reduction of checked rows for SPD with respect to SpiralPDE with partial distortion comparison every 1,2,4,8,16 pixels, for all tested sequences

Table 1. Average checked rows for various algorithms with respect to SpiralPDE. P2 is the "Proposed 2" in [15]. Data for all algorithms are for these five sequences: "carphone", "claire", "foreman", "grandmother", "trevor", also used by [15]. (*) SPD does comparisons every 8 pixels (equivalent to a half row) as discussed in Section 5.

SEA	PDE	SpiralPDE	P2	SPD
3.38	7.62	3.62	3.06	2.93*

6 Conclusions and Future Work

In this paper we have presented a new full-search (lossless), fast-matching, block motion estimation algorithm, which uses the partial distortion elimination approach; this algorithm is denoted as Sobol Partial Distortion (SPD).

It uses a new matching strategy, based on Sobol's sequence instead of row by row top to bottom scanning order, to quickly compute minimum distortion, and reduce block matching computation. No image-dependent computations are required.

The proposed algorithm performs well in comparison with other existing full search algorithms. In particular our experimental evaluations have reported an

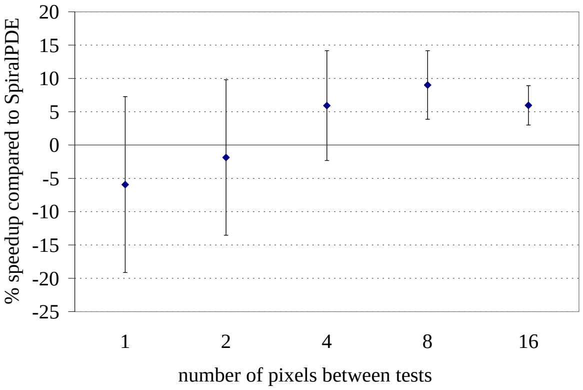

Fig. 3. Average value and standard deviation of the percentage of reduction in total encoding cpu time for SPD with respect to SpiralPDE with partial distortion comparison every 1,2,4,8,16 pixels, for all tested sequences

average computational gain of up to 20% with respect to SpiralPDE algorithm, as reported in Table 1. The proposed algorithm is independent of methods based on SEA and can be used with them to improve global speedup.

The results which we have obtained are a stimulus for further exploration of the proposed methodology in the light of improved performances and of fast special purpose architecture. Future work aims to further improve the matching strategy. The uniformity of Sobol's sequence could also be used in a lossy method based on pixel decimation in the matching phase.

References

1. Dufaux, F., Moscheni, F.: Motion estimation techniques for digital TV: A review and a new contribution. Proceedings of the IEEE, **83** 6 (1995) 858–876
2. Stiller, C., Konrad, J.: Estimating motion in image sequences. IEEE Signal Processing Magazine **16** 4 (1999) 70–91
3. Po, L.M., Ma, W.C.: A novel four-step search algorithm for fast block motion estimation. IEEE Trans. On Circuits and Syst. for Video Technol. **6** 3 (1996) 313–317
4. Lu, J., Liou, M.L.: A simple and efficient search algorithm for block-matching motion estimation. IEEE Trans. On Circuits and Syst. for Video Technol. **7** 2 (1997) 429–433
5. Cheng, F.H., Sun, S.N.: New fast and efficient two-step search algorithm for block motion estimation. IEEE Trans. On Circuits and Syst. for Video Technol. **9** 7 (1999) 977–983
6. Wang, Y., Wang, Y., Kuroda, H.: A globally adaptive pixel-decimation algorithm for block-motion estimation. IEEE Trans. On Circuits and Syst. for Video Technol. **10** 6 (2000) 1006–1011

7. Wang, H.S., Mersereau, R.M.: Fast algorithms for the estimation of motion vectors. IEEE Trans. On Image Processing **8** 3 (1999) 435–438
8. Cheung, C.K., Po, L.M.: Normalized partial distortion search algorithm for block motion estimation. IEEE Trans. On Circuits and Syst. for Video Technol. **10** 3 (2000) 417–422
9. Lengwehasatit, K., Ortega, A.: Probabilistic partial-distance fast matching algorithms for motion estimation. IEEE Trans. On Circuits and Syst. for Video Technol. **11** 2 (2001) 139–152
10. Li, W., Salari, E.: Successive elimination algorithm for motion estimation. IEEE Trans. On Image Processing **4** 1 (1995) 105–107
11. Oh, T.M., Kim, Y.R., Hong, W.G., Ko, S.J.: A fast full search motion estimation algorithm using the sum of partial norms. Proc. ICCE 13–15 June (2000) 236–237
12. Coban, M.Z., Mersereau, R.M.: A fast exhaustive search algorithm for rate-constrained motion estimation. IEEE Trans. On Image Processing **7** 5 (1998) 769–773
13. Eckart, S., Fogg, C.: ISO/IEC MPEG-2 software video codec. Proc. SPIE Vol. 2419 (1995) 100–118
14. Telenor R&D, ITU-T recommendation H.263 software implementation, Digital Video Coding Group (1995)
15. Kim, J.N., Choi, T.S.: A fast full-search motion-estimation algorithm using representative pixels and adaptive matching scan. IEEE Trans. On Circuits and Syst. for Video Technol. **10** 7 (2000) 1040–1048
16. Brünig, M., Niehsen, W.: Fast full-search block matching. IEEE Trans. On Circuits and Syst. for Video Technol. **11** 2 (2001) 241–247
17. Sobol, I.M.: On the distribution of points in a cube and the approximate evaluation of integrals. USSR Computational Mathematics And Mathematical Physics **7** 4 (1967) 86–112
18. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: Numerical Recipes In C: The Art Of Scientific Computing. 2nd edn. Cambridge University Press (1993)