Signal Processing: *Image Communication* 31 (2015) 25–33

Contents lists available at ScienceDirect

Signal Processing: *Image Communication*

journal homepage: www.elsevier.com/locate/image

A fast partial distortion search algorithm for motion estimation based on the multi-traps assumption

Xiao-Peng Xia [a,*], En-Hai Liu [a], Jun-Ju Qin [b]

[a] Institute of Optics and Electronics, Chinese Academy of Sciences, Chengdu City 610209, Sichuan Province, PR China
[b] Chengdu Normal University, Chengdu 610072, China

ARTICLE INFO

Article history:
Received 20 July 2014
Received in revised form
28 November 2014
Accepted 28 November 2014
Available online 5 December 2014

Keywords:
Partial distortion
Motion estimation
Multi-traps
Complexity calculation

ABSTRACT

Full search has the best matching accuracy but costs the most time, while fast search algorithms can achieve a high speed but are easy to be trapped in local minimums. To compensate the shortcomings of the existing algorithms, this paper proposes a fast partial distortion motion estimation algorithm based on the multi-traps assumption (MT-PDS). It mainly consists of three steps: (1) estimate the number of traps in the search area, (2) obtain the positions of the traps by using the k-th (k=0,1,2,…15) partial distortion, which contributes most to the true sum of absolute difference (SAD), to perform the coarse search, and (3) get the positions of the deepest traps and search around them to get the global minimum. Besides, the proposed algorithm also introduces an adaptive search method and a sparse search pattern, which further reduce the computations. Experimental results show that the proposed MT-PDS is about 160 times faster than the full search on average; and the speed-up can achieve over 180 times for the low motion contents. What is more, it only degrades the quality by −0.0178 dB and slightly increases the bit rate by 0.735%, which can be considered ignorable. Those advantages make the MT-PDS a very useful tool in real time applications, such as video compression, pattern recognition, target tracking, etc.

1. Introduction

Block matching motion estimation is widely used in many fields such as video compression, pattern recognition, etc. So far, many block matching algorithms have been proposed.

The full search (FS) provides the best quality, but it suffers from the burden of high complexity. Hence, researchers have proposed many fast search algorithms such as the three-step search (TSS) [1], four-step search (FSS) [2], diamond search (DS) [3], the hexagon-based search algorithm (HEXBS) [4], etc. These algorithms are based on the assumption that the global minimum motion vector distribution is center-biased. Decreasing the computational complexity greatly, the phenomenon of being trapped in local minimums can hardly be avoided, which leads to a relatively high degradation of quality.

For the purpose of avoiding being trapped in local minimums, some lossless fast search algorithms are proposed [5–7]. Partial distortion search (PDS) [7] is a good example. PDS uses the accumulated partial SAD to eliminate the impossible MV candidates. In the later research, it is found that the different pixels in the MB have different contributions to the true SAD. So sorting-based partial distortion algorithms [8–12] are proposed. Although the lossless fast partial distortion search algorithms do not introduce any quality degradation, they can only improve the speed to a limited degree.

Furthermore, in order to get higher search speed while allowing slight quality degradation, partial distortion search algorithms are also used in the lossy search [13–21], such as the

* Correspondence to: Institute of Optics and Electronics, Chinese Academy of Sciences, Mailbox 350, Chengdu City 610209, Sichuan Province, PR China. Tel.: +86 28 85101400; fax: +86 28 85865157.
E-mail addresses: tabam07uestc@163.com (X.-P. Xia), leh07@ioe.ac.cn (E.-H. Liu), 745977895@163.com (J.-J. Qin).

http://dx.doi.org/10.1016/j.image.2014.11.007
0923-5965/

normalized partial distortion search algorithm (NPSD) [16], the two-step edge based partial distortion search algorithm (TS-EPDS) [20] and its further work, the adaptive two-step edge-based partial distortion search algorithm (ATS-EPDS) [21], etc. They yield a speed-up much more than the lossless partial distortion algorithms by using fast skipping and early termination strategy.

Lately, Chen et al. have proposed a new adaptive MB-mean difference sorting scheme based NPDS algorithm (AMDNPDS) [22], which reduces the check pixels by about 11.02 times versus typical PDS algorithm. In addition, Nijad et al. [23] combine the hierarchical search (HS) algorithm with the cross-diamond search pattern and the enhanced three-step-search algorithm (ETSS), getting an 83.4% of complexity reduction versus FS and a matching quality over 98%.

Although the lossy fast partial distortion algorithms have greatly improved the speed with relatively low quality degradation, they can still be improved.

This paper presents a new fast partial distortion algorithm based on the multi-traps assumption (MT-PDS). It assumes that there are traps in the search area and the global minimum is within the positions around the deepest traps. To find the deepest traps, a few steps are taken. Firstly, calculate the complexity of each sub-MB and estimate the number of candidate traps. Then, find the candidate traps by using the sub-MB, which can best represent the current MB, to perform the coarse search. Further, search the positions around the chosen traps and get the deepest trap. Lastly, the DS search with the step of 1 is applied to the pixels around the deepest trap to find the precise location of the best MV. Here, it is necessary to point out that the "MB" appearing in this paper denotes the PU block (16×16 pixels), and it is not the same as the "MB" in HEVC.

The rest of this paper is organized as follows. Section 2 introduces the proposed MT-PDS algorithm. The experimental results and discussions are presented in Section 3. Finally, Section 4 concludes the paper.

2. Proposed MT-PDS algorithm

2.1. Multi-traps assumption

Before illustrating the concept of traps, the definition of the local minimum area, which is not totally the same as the traditional meaning, is firstly defined here. In the local minimum area, the SAD of the center position is smaller than (or equal to) that of other positions within the area, which is shown as follows:

$$
\begin{aligned}
&Local\_minumum\_area = \{(x,y)_0, (x,y)_1, \ldots (x,y)_i\};\\
&\quad SAD(x_i, y_i) <= SAD(x,y)_i \quad \text{for:}\\
&\quad \forall x \in (x_i - R_i, x_i + R_i);\\
&\quad \forall y \in (y_i - R_i, y_i + R_i)
\end{aligned} \tag{1}
$$

where (x_i, y_i) is the position of the local minimum, which is also the center of the area $(x, y)_i$; R_i is the radius of the i-th local minimum area; $(x, y)_i$ is the random position within the i-*th* local minimum area; i is the index of the current local minimum area.

From the definition in Eq. (1), it can be inferred that there could be many local minimum areas in the search area, especially when R_i is relatively low and the search range is high. For an analogy, the local minimum areas can be taken as the hollows in the road. The number of them could be high and it is not easy to find every hollow. So the traditional fast search methods such as TSS, FSS are easy to be trapped in the local minimums. Besides, the number of local minimum areas in the search area also has an effect on the search speed of PDS algorithms. In the positions within local minimum areas, the PDS algorithms always need to calculate more times of partial distortions than in other positions, which results in great computations.

Since local minimum areas may be in great quantity and are hard to be found, the concept of traps is introduced here. The traps, which are similar with the local minimum areas, are defined as follows:

$$
\begin{aligned}
&Traps = \{(x,y)\};\\
&\quad (x,y) \in Local\_minumum\_area((x,y)'_0, (x,y)'_1, \ldots (x,y)'_l);\\
&\quad \text{where } l <= i;\\
&\quad Local\_minumum\_area((x,y)'_0, (x,y)'_1, \ldots (x,y)'_i) =\\
&\quad Sort(Local\_minumum\_area((x,y)_0,\\
&\quad (x,y)_1, \ldots, (x,y)_i), descent);
\end{aligned} \tag{2}
$$

where the function of $Sort()$ is to sort the local minimum areas ($Local\_minumum\_area$) in descending order of the SAD in the center positions. Namely, the traps are parts of the local minimum areas whose minimums are the smallest. For an analogy, traps can be taken as the deep well in the ground. The number of traps is much smaller and the traps are easier to be positioned than the hollows.

Fig. 1 shows the SAD distributions of the central 5 by 5 pixels of the bus sequence in the reference frame. Blue block area indicates that its SAD is relatively low, while red block area indicates that its SAD is high. From Fig. 1(a) it can be seen that there are local minimum areas even in the areas with relatively high SAD. However, the traps (when $l=6$ in Eq. (2)) which have been marked by the red circles shown in Fig. 1(b) are much fewer.

Hence, the block matching problem is converted to the problem of finding the deepest trap in the search area.

2.2. The search based on multi-traps

In order to get the positions of traps with low-cost computations, some analysis is made as follows:

Let d_k be the k-*th* partial SAD, then the k-th accumulated partial SAD (SAD_{kth}) can be

$$SAD_{kth} = \sum_{m=0}^{k} d_m \quad k = 0, 1, \ldots, 15 \tag{3}$$

Experiments show that different pixels in the MB contribute to the actual SAD differently, especially when the pixels are in high activities, such as edges and texture [24,25]. So the d_k ($k=0,1,\ldots 15$) in descending order can be obtained by sorting the pixels in the MB, namely, $d_k >= d_{k+1}$ for $k=0,1\ldots,14$. Then the true SAD can be described as follows:

$$SAD = \sum_{i=0}^{i=15} d_i = \sum_{i=0}^{i=15} c_i \times d_0 = d_0 \times \sum_{i=0}^{i=15} c_i \tag{4}$$

where c_i is the weight factor and $c_i = d_i/d_0$. It is straight that