# NON-OVERLAPPING SEARCH PATTERN FOR LOGARITHMIC SEARCH MOTION ESTIMATION

Astrid Lundmark, Linköping University, Sweden, astrid@isy.liu.se

**Abstract:** *Logarithmic search motion estimation is popular in video codecs because of its low complexity, and because it finds the motion vector in a pre-defined number of computation steps. Motion estimation still remains the most demanding part of the encoding scheme, so further improvements in terms of reduced computational complexity are desirable. Motion estimation is also rewarding, in terms of bit-rate reduction, so the quality of the resulting motion vector field has high influence on video coding performance. In this paper we will show how the computational complexity can be lowered and the quality of the motion vector field improved by a simple change of the search pattern.*

**Key words:** *motion estimation, logarithmic search, three-step search, search pattern, search area, video coding*

## 1. INTRODUCTION

In video coding, the current frame is divided into blocks, which are predicted from the previously encoded frame. Standard codecs use motion compensated prediction with the same motion vector for the entire block. To find the optimal motion vector, given enough time, all decodable motion vectors could be tried for generating the prediction and encoding the prediction error. The motion vector generating the shortest bit stream at a pre-defined allowed distortion should be chosen. The above procedure is however too time consuming for real-time processing, therefore two types of simplifications are normally made, namely **simplified evaluation criterion** and **reduced search**.

A simplified evaluation criterion is the sum of absolute distances (SAD)

$$SAD = \sum_{block} |I(x, y, t) - \hat{I}(x - u, y - v, t - 1)| \quad (1)$$

where $I$ denotes image intensity and $\hat{I}$ decodeded image intensity, the summation is made over image coordinates $x$ and $y$ belonging to the block to encode, $u$ and $v$ constitutes the motion vector to be evaluated, and $t$ denotes the time index of the frame. To further reduce the computational complexity, the summation could be made over a subset of the $x, y$ locations in the block [1].

Reduced search strategies rely on the evaluation criterion to vary smoothly with spatial location.

Hierarchical search operates in an image pyramid, where the original image constitutes the base layer of the pyramid. The algorithm starts at a high level in the pyramid. A small search area, counted in pixels at this resolution, corresponds to a large area at base layer resolution. At each level of the pyramid, the motion vectors coming from the layer above are used as initial estimates and refined.

Table 1: The longest motion vectors in full pixel resolution possible to reach by $i$ iterations of logarithmic motion estimation using the search pattern in Fig. 1 (a) ($l_2$) and the proposed search pattern in Fig. 1 (b) ($l_3$).

| $i$ | $l_2$ | $l_3$ |
|---|---|---|
| 1 | 1 | 1 |
| 2 | 3 | 4 |
| 3 | 7 | 13 |
| 4 | 15 | 40 |
| 5 | 31 | 121 |

Logarithmic search [2] and the closely related 3-step search [3] are similar to hierarchical search in the sense that first a coarse estimate is produced, which is then stepwise refined. Unlike hierarchical methods, logarithmic search methods operate in image resolution. Fig. 1 (a) shows how the search pattern looks if a 3 by 3 search window is used in each step, and the size of the refinement vector is reduced by a factor of 2 between the iterations. From Fig. 1 (a) it is clear that many of the arrows meet, i. e. the search windows overlap. This can at first glance seem to be an advantage, since it allows some error recovery. However, the overlapping search windows imply a strange a priori probability distribution assumption for the motion vectors leading to sub-optimal decisions. Overlapping search windows also make the computational load unnecessarily high. Error recovery is handled well by using multiple search paths [4, 5].

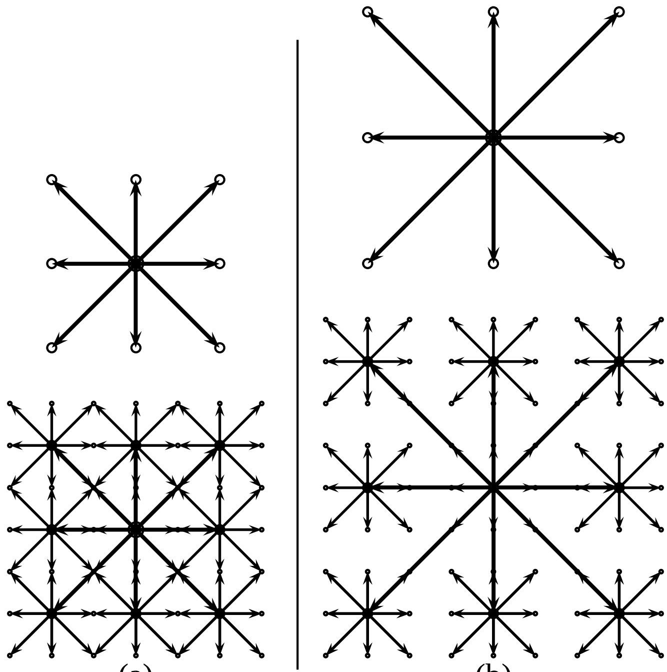
Multiple search paths [4, 5] means keeping not only the best candidate motion vector at each iteration, but a pre-defined number of motion vectors. This improves performance at the expense of increased computational complexity.

## 2. NON-OVERLAPPING SEARCH PATTERN

We propose to use non-overlapping search fields for logarithmic motion estimation. This can be achieved by using the search pattern shown in Fig. 1 (b). The non-overlapping search fields have the advantages of being able to find longer motion vectors in fewer iterations, see Table 1, and to allow easy and effective handling of multiple search paths.

## 3. EXPERIMENTAL RESULTS

We used the complex-motion scene flowergarden to compare the performance of motion-compensated prediction using full search, traditional logarithmic search, and the proposed logarithmic search. Four iterations were made in both the logarithmic estimators. For the traditional search scheme, this gives a natural limitation in the length of motion vectors to 15. For the proposed scheme, we limited the search area to the same size, in order to be able to make fair comparisons. For the logarithmic search methods, the number of match value calculations was $36s$, where $s$ stands for number of search paths. This number could be lowered if using the fact that some of the matches have already been calculated. That procedure, however, requires some overhead, and it is doubtful if it is meaningful when the match criterion is relatively simple. For full search, the search area was also 31 by 31 pixels, which gives 961 match



(a) (b)

Figure 1: (a): Two iterations of logarithmic matching with half as large refinement vectors in each step and a search window of 3 by 3. In the first iteration (top image), 9 positions are searched. Around the best match(es) found, a refined search is made (bottom image). All possible outcomes of the algorithm are shown. (b:) The proposed logarithmic matching using refinement vectors with lengths $\frac{1}{3}$ of the sizes in the previous step, and a search window of 3 by 3. All possible outcomes of the algorithm are indicated to show that it is possible to find all motion vectors.

value calculations. The outcome in terms of PSNR for the motion compensated prediction is shown in Fig. 2

We see that motion compensation is really paying off, since using the previous picture without motion compensation gives a PSNR of only 15.7 dB, while using full search motion estimation gives a PSNR of 25.5 dB. The basic versions of logarithmic search using only one search path give PSNR around 20 dB (20.0 for the traditional search pattern and 19.9 for the proposed). Using more search paths improves PSNR substantially for the logarithmic motion estimation algorithms. The computational complexity increases linearly with the number of search paths. For 9 search paths, the computational complexity for logarithmic search is a third of the computational complexity for full search. The non-overlapping scheme has capacity for finding motion vectors up to a length of 40 with this number of iterations. If this capacity would be used, full search would need twenty times more match value calculations than logarithmic search with 9 search paths. For larger motion vectors, the relative difference between the required computational complexity for full search and logarithmic search increases, since the number of match value calculations rise with square of the motion vector length, while the number of matches for logarithmic motion estimation rises with the logarithm of the motion vector length.

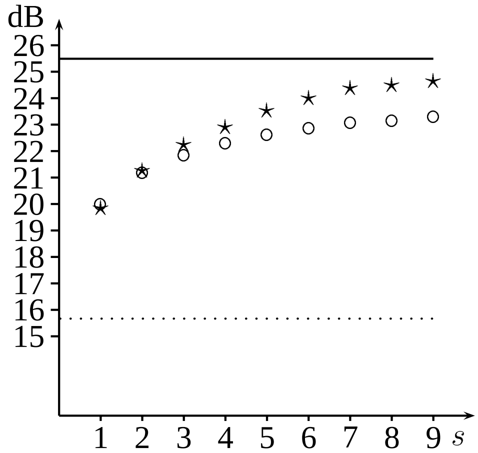



Figure 2: Results of motion compensated prediction using no motion estimation (dotted line), full search motion estimation (solid line), logarithmic estimation with traditional search pattern (○), and logarithmic motion estimation with the proposed search pattern (⋆), as a function of the number of search paths, $s$. Test sequence was “flower garden” in 352 by 240 pixels resolution.

### 4. CONCLUSIONS

The proposed non-overlapping search scheme for logarithmic motion estimation shows improvements in computational complexity and proposes a more simple and straightforward alternative to the partly-overlapping schemes that have been presented so far. It is ideally suited for use with multiple search paths.

## References

[1] André Zaccarin and Bede Liu. Fast algorithms for motion estimation. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume III, pages 449–452, March 1992.

[2] J.R. Jain and A.K. Jain. Displacement measurement and its application in interframe image coding. *IEEE Transactions on Communications*, COM-29:1799–1808, December 1981.

[3] T. Koga, A. Iinuma, Y. Iijma, and T. Ishiguro. Motion-compensated interframe coding for video conferencing. In *Proceedings of the National Telecommunications Conference*, pages G5.3.1–G5.3.5, 1981.

[4] Anders Lindman. Implementering och utvärdering av rörelseestimeringsalgoritmer för kodning av TV/HDTV. Master’s thesis, Linköping University, Sweden, February 1992.

[5] K. W. Cheng and S. C. Chan. Fast block matching algorithms for motion estimation. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2311–2314, 1996.

In full search motion estimation entire frame as a search window size.Consider a block of N A— N pixels from the candidates from at the coordinate position (r, s) as shown and then consider a search window having a range Â±w in both directions in the reference frame, as shown. For each of the (2w+1) ^2 search positions (including the current row and current column of the reference frame), the candidate block is compared with a block of size N Ã— N pixels according to one of the best matching criteria is discussed and the best matching block, along with the motion vector is determined only after a... 24. Motion Estimation â€“ Fast Search Methods. Three-Step Search (TSS). â€¢ For typical applications of 2D Logarithmic search in. broadcast TV (480 720 pixels, 30 fps) for N1 = N2 = 16 and p = 7, 1.02 GOPS required. â€¢ TSS usually underperforms full search by 0.5 to 1 dB in PSNR for common test video sequences.Â  â€’ Equidistant search patterns. â€’ May skip global minimum in low motion cases. Copyright 2007 by Lina J. Karam. 25. Motion Estimation â€“ Fast Search Methods. Efficient Three-Step Search (ETTS) (Jing & Chau [2004]). â€¢ More emphasis on center biased motion. search window and of the associated search strategy are critical for a motion estimation algorithm. Figure 8 - Left: the current image with the block Bp,q highlighted. Right: the reference image.Â  When again a zero motion vector is found, either the estimation is stopped, or a third step can be envisaged, involving a search in the direction given by the original estimation point and the previous estimation. Note however that this step is optional and it can be skipped if a limited-complexity algorithm is targeted. Motion estimation is the process of determining motion vectors that describe the transformation from one 2D image to another; usually from adjacent frames in a video sequence. It is an ill-posed problem as the motion is in three dimensions but the images are a projection of the 3D scene onto a 2D plane. The motion vectors may relate to the whole image (global motion estimation) or specific parts, such as rectangular blocks, arbitrary shaped patches or even per pixel. The motion vectors may be...