New Predicted Spiral Search Block Matching Algorithm - PSSBMA

Jan PÍKA
Department of Telecommunications
Czech Technical University in Prague
Technická 2, 120 00 Praha
The Czech Republic

Abstract

This article describes the modification of the full search algorithm ESBMA (Exhaustive Search Block Matching Algorithm), which leads up to 40% speed increase. The modification is based on the ESBMA motion field analysis results. The major modifications to the ESBMA are:

- *Introduction of sub-optimality by thresholding the matching criterion (MAEthr);*
- *Respecting constraints on motion vectors resulting from „head and shoulder" scenes by changing the position of the search start;*
- *Respecting the dependence of motion vectors (MV) by prediction introduction.*

Keywords

Very low bit rate compression, motion vector estimation, motion vector prediction

1. Introduction

There are many major standards for very low bit rate (VLBR) compression of video signals; e.g. ITU-T H.261 - Video codec for audiovisual services for rates $p \times 64$ kbps [1], group of recommendations ITU-T H.324 – Terminal for low-speed multimedia communication, the part of which is recommendation ITU-T H.263 - Video codec [1], or MPEG-4, video part of which is ISO 14496-2 [4]-[5].

These standards are based on the first generation compression techniques, which use block-based frame processing (this holds for MPEG-4 if VLBR is the target application). Second generation compression techniques use the segmentation of frames into logical entities and are used in MPEG-4 systems at rates above VLBR [1] – [5].

With respect to the limited computational power of the computers and telecommunication systems, the key to the successful implementation of systems for creation, transmission and presentation of video signals is the effective solution of video signal compression and decompression [1]. The most computational power demanding operation is motion vectors estimation and a discrete cosine transform (DCT) [1].

Therefore author's research was focused on the enhancement of motion vector estimation techniques. As a block-matching criterion the minimum of *MAE*(*i*,*j*) function was used, which is the most widely used method [1], [9]. This criterion compares a macro-block *MB*(*i*,*j*) – *i*-th column and *j*-th row – of the frame at the time *t* with the area of the same size in the reference frame at the time *t*+*x*. MAE function is computationally intensive and exhibits the same performance as the correlation function or the mean square error [1]

$$MAE(i,j)=\frac{1}{MN}\sum_{k=0}^{M-1}\sum_{l=0}^{N-1}\left|C(x+k,y+k)-\right.$$
$$\left.-R(x+k+i,y+l+j)\right|$$

for

$$\{-p \le i,j \le +p\} \tag{1}$$

where *M*, *N* is *MB* size, *C*(*x*,*y*) are pixels of the actual *MB* and *R(x,y)* are pixels of the reference *MB*.

The contribution of this research is two-fold. Firstly it demonstrates the characteristics of head-and-shoulder scene motion field; secondly it introduces the prediction of MV as a mean to speed up MV search algorithm and points out of a direction for further research in this field.

Chapter 2 introduces results of the research that has been conducted in this field. Chapter 3 deals with the motion field analysis of head-and-shoulder video and highlights its major characteristics, which enable us to remarkably enhance the MV search process. Further on it describes the resulting algorithm PSSBMA, which is the modification of ESBMA, including drawbacks associated with its deployment. In chapter IV temporal performance and resulting motion field are presented. Chapter V discusses the performance PSSBMA and its implication for practical deployment. The direction for further research is suggested in chapter 4. The last chapter present references.

2. Summary of related research

Recently, many interesting MV search algorithms for VLBR applications have been developed. Majority of algorithms use the minimum of *MAE*(*i*,*j*) function as block matching criterion. More sparsely criteria such as normalized cross-correlation function NCCF [11], statistical correlation [12], mean square error [1] or number of pixels classified as matching according PDC algorithm [13] are used.

The fundamental MV search algorithm is exhaustive search – ESBMA. The algorithm calculates the *MAE*(*i*,*j*) for all search positions in the search window and then determines minimum of *MAE*(*i*,*j*). In search window it finds global minimum and hence from minimum of *MAE*(*i*,*j*) it has optimal performance (the same holds for the quality of reconstructed frame before adding a difference frame), however from motion field view point it is by no means an optimal algorithm as it accommodates no means of generating a smooth motion filed resembling natural motion field of head-and-shoulders scenes. The non-smooth motion field deteriorates the compression ratio of MV differential coding [1].

Zeng and Liou developed a three-step-search algorithm – TSS, which calculates the value of *MAE*(*i*,*j*) only for three positions in the search window. This results into a very fast performance. The drawback is the high entropy of difference frame and increased bandwidth needed for transmission [1], [8]. The principle of setting a fixed number of search steps and determining a proper strategy for selecting search positions in the search window is used by a number of other search algorithms [1], such as 2D logarithmic search or parallel hierarchical one-dimensional search [1].

Pickering, Arnold and Frater suggested a MV estimation method that employees an adaptive number of search steps - Adaptive Search Length Algorithm – ASL, which varies based on characteristics of each macro-block, however it maintains the average number of search steps per *MB* the same for each frame. Compared to ESBMA, ASL yields a 0.25 dB lower quality at 90% decrease of *MAE*(*i*,*j*) calculations [9]. As ASL searches a limited number of positions it is classified as sub-optimal algorithm.

From optimality viewpoint, an interesting algorithm has been developed by Yih-Chuan Lin and Shen-Chuan Tai [10]. Instead of calculating *MAE*(*i*,*j*) for all positions in the search window, it calculates a more simple function based on integral projection of pixels. Based on this criterion the number of possible candidates gets remarkably reduced. Only then, the *MAE*(*i*,*j*) function is applied on the reduced set of candidate positions. This algorithm can therefore be classified as optimal.

3. PSSBMA Algorithm

This algorithm is based on two fundamental theses:

A) In most VLBR applications of „head-and-shoulders“ type the motion is limited and values of MV are low [1];

B) Moving macro-blocks are part of a larger logical entity (speaker’s head, whole body, etc.) and their MVs are not independent of their neighbouring MBs.

In order to support thesis B, motion analysis is performed.

3.1 Motion Analysis

The goal of analysis is the comparison of ESBMA, which does not respect dependencies among neighboring *MB*s with real motion characteristics. The analysis was performed in Matlab 5.2 environment, using a video sample created at the Dept. of Telecommunications at FEL CVUT. The video sample was recorded with a single-chip CCD Panasonic NV-MS4 PAL camera. After recording on tape the sample was converted into AVI format using AuthorWare 3.5. For further ease of processing the video sample was decomposed into particular frames:

Number of frames:	51
Frame format:	BMP, RGB, 3×8 bit/pixel
Resolution:	CIF (352×288 pixels)
Frame rate:	25 fps

Performance of the algorithms was measured on a Windows NT Workstation with 128 MB RAM/ 200 MHz. Process priority was set to *normal*.

Different frame rates needed during experiments were achieved by evenly bypassing relevant intermediate frames. The numbering of frames used in figures below refers to 25 fps frame rate.

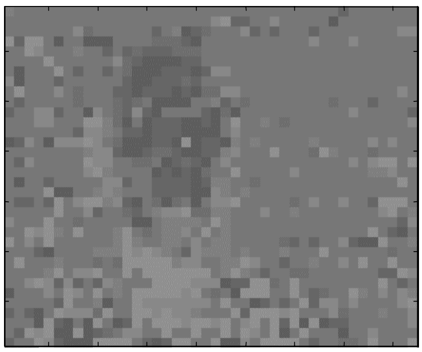
As H.261 and H.263 standards use YC_rC_b format [1], the frames were converted from RGB into YC_rC_b prior to the motion analysis according to equations (2) [1]

$$\begin{aligned} Y &= 0.299(R-G) + G + 0.114(B-G) \\ C_b &= 0.564(B-Y) \\ C_r &= 0.713(R-Y) \end{aligned} \tag{2}$$

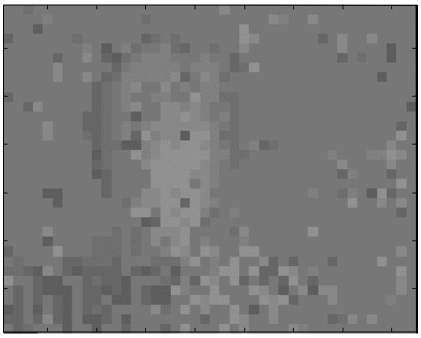
where Y represents hue and C_b, C_r are chrominance components.

As mentioned above, ESBMA searches all position in the search window before it determines the minimum of *MAE*(*i*,*j*) function. Therefore it does not matter what is the order of the search positions during the search.

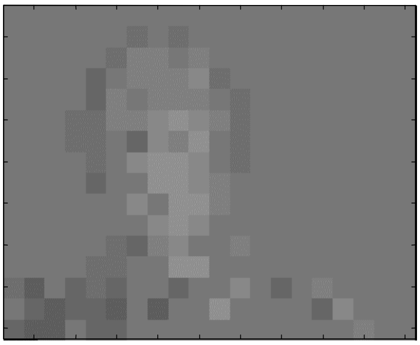
X and Y components of the determined MVs are depicted in Fig. 1 for a search window size equal to 3. Fig. 1 also shows X and Y components of MV determined for blocks of 8×8 pixels.



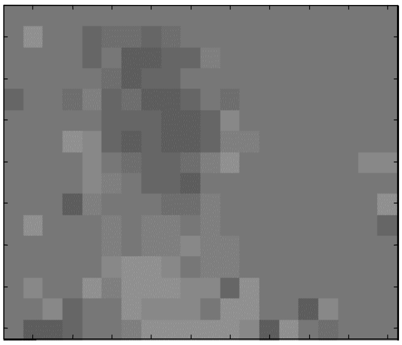
(a)



(b)



(c)



(d)

Fig. 1 X and Y components of MVs for ESBMA: frame 18, 6.25 fps, window 3; (a) x-components for block 8×8; (b) y-components for block 8×8; (c) x-components for block 16×16; (d) y-components for block 16×16.

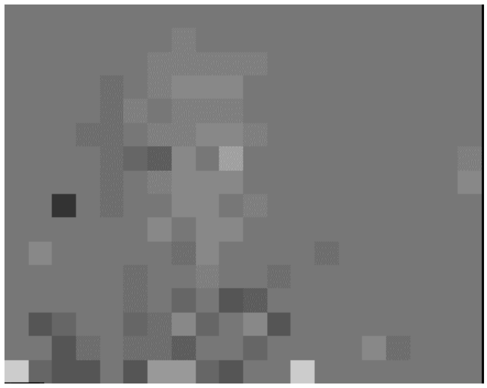
To better perceive the spatial correspondence of MVs with the scene in a frame, the *X* and *Y* components are visually presented as grayscale values. Such representation is convenient for overall assessment of the motion field smoothness and visually highlights the spatial dependence of MVs of neighboring *MB*s. Zero value (no motion) is expressed by gray color present in the upper right-hand corner of Fig. 1c. Lighter blocks represent positive values and darker ones negative values of *X* and *Y* components of MVs.

Despite the scene having no motion in the background the search detects some motion, the amount of that grows with smaller size of blocks. More motion is detected for 8×8 blocks than for 16×16 *MBs*. Let‘s call this motion *false motion*. False motion disturbs smoothness of the motion field, and increases the bandwidth for transmitting MVs.

This phenomenon can be explained by camera noise, fluctuations of scene illumination, PAL system distortion, and noise introduced by recording on a tape and by conversion into AVI format. For larger sizes (16×16) this noise is better averaged out.

The analysis also shows that the smoothness depends in the size of the search window. With larger search windows the probability of detecting a false motion with a very large MV grows. Such a situation is illustrated in Fig. 2b where we can see several large X components.

(a)



(b)

Fig. 2 The x-components of MVs for frame 17 at 6.25 fps; ESBMA search; (a) window 3; (b) window 10.

The motion field analysis can be concluded by following statements:

- Both Fig. 1 and Fig. 2 show that most MVs are similar to those of their neighbors, which supports thesis B.
- Decreasing block size and growing search window result into growing amount of false motion detected.

3.2 PSSBMA

With regards to the conclusions drawn form the motion analysis, several modifications to ESBMA were proposed. To better demonstrate their effects, modifications are applied step by step. Each modification results to a new algorithm name and includes all previous modifications.

The first change is the introduction of sub optimality by stepping away from searching the minimum of $MAE(i,j)$ and setting a threshold value MAE_{thr} as a criterion for block matching. This results into **TSBMA** (*Threshold Search Block Matching Algorithm*). In the experiments performed, MAE_{thr} was set to 10. The order of search position stays the same as for ESBMA with the first one in the upper left-hand corner of the search window and the last in the lower right-hand corner. Therefore the first values of $MAE(i,j)$ are calculated for position with lower probability of meeting the block-matching criterion thus not respecting thesis A. Therefore the next modification changes the order of search window walk-through.

SSBMA (*Spiral Search Block Matching Algorithm*) respects thesis A and does the first calculations of $MAE(i,j)$ at positions representing minimum values of MVs (MV_x= =MV_y=0). The search window walk-through follows the decrease of probability of finding a MB, which meets MAE_{thr} criterion and is depicted in Fig. 3. The remaining weak spot of the algorithm is an unutilized spatial dependency of MVs.

The last enhancement utilizes the temporal dependency of the MVs and is called **PSSBMA** (*Predicted Spiral Search Block Matching Algorithm*). New feature is MV prediction where a predictor is the MV of the previous *MB*. There are also other ways of determining the predictor such as the average of MVs of the *MB*s surrounding the current *MB*. The resulting algorithm eventually contains the following changes applied to ESBMA:

- Sub-optimality by introducing *MAEthr*;
- Respecting limited values of MVs for „head-and-shoulders" scenes;
- Utilization of mutual dependency of MVs through their prediction.

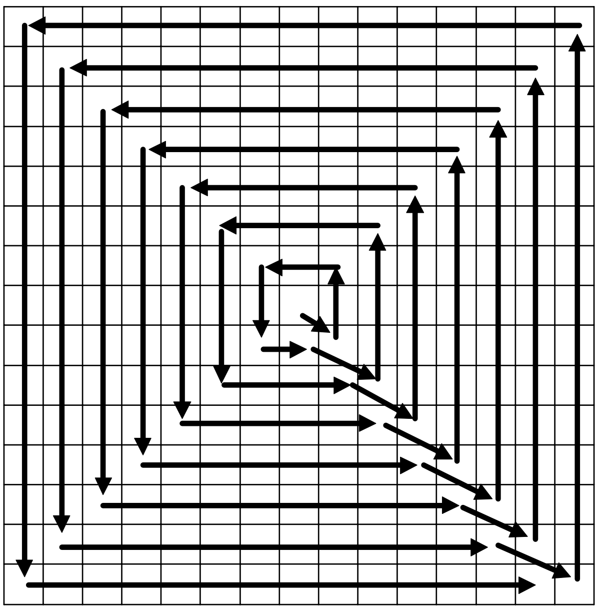


Fig. 3 Search window walk-through along the search spiral

A drawback associated with the prediction is its growth demonstrated in table 1. MV of the *MB* is searched for in the search window of 15 pixels. In the worst case for all subsequent *MB*s the matching criterion may result into estimating the same MVs all pointing to one of the corners of the search window (for instance MV=[15,15]). As each *MB* is the predictor for the following one, the MV for each *MB* with respect to idle status without prediction (MV') grows excessively. Therefore a prediction magnifies the *false motion* phenomenon.

The above-described situation happens very rarely and depends on the scene characteristics. Despite this it is necessary to secure the algorithm against unrealistically high MV values. During experiments the MV estimated by PSSBMA was, for properly set MAE_{thr}, equal to the ones determined by ESBMA for the majority of *MB*s.

MB	PR	MV'	MV
1	(0,0)	(15,15)	(15,15)
2	(15,15)	(15,15)	(30,30)
3	(30,30)	(15,15)	(45,45)
4	(45,45)	(15,15)	(60,60)

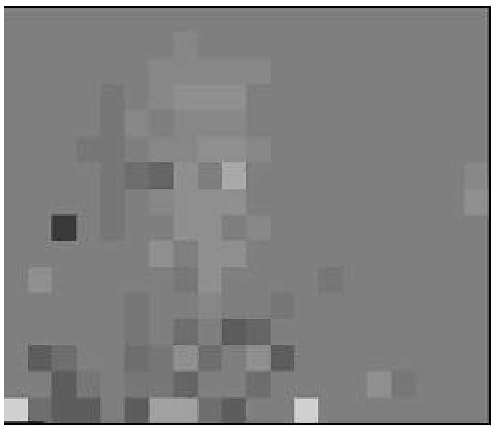
Tab. 1 Growth of MV predictor

PSSBMA employs the easiest predictor growth control mechanism, which zeroes the predictor if it violates the frame boundary.

4. Performance of the Algorithms

Motion fields presented on the following pictures, temporal performance table and distribution functions of time needed for determining a MV create a baseline for discussion in chapter 5.

Temporal performance of the algorithms discussed was measured by time functions of MATLAB 5.2. Time t_m is a time needed for determining the MV of a single *MB* and is averaged over the whole video sample.

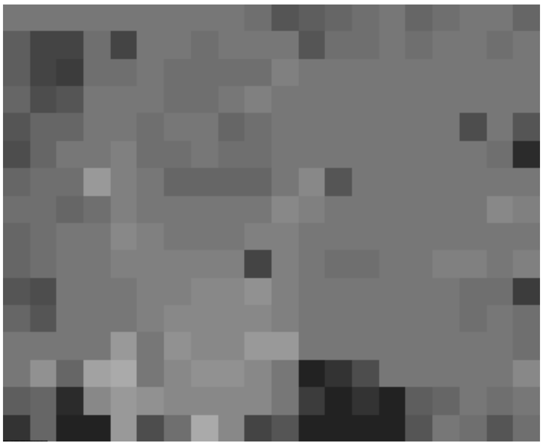


(a)

(b)

Fig. 4 Motion field of frame 17; ESBMA, 6.25 fps, window 10; (a) x-component; (b) y-component

(a)



(b)

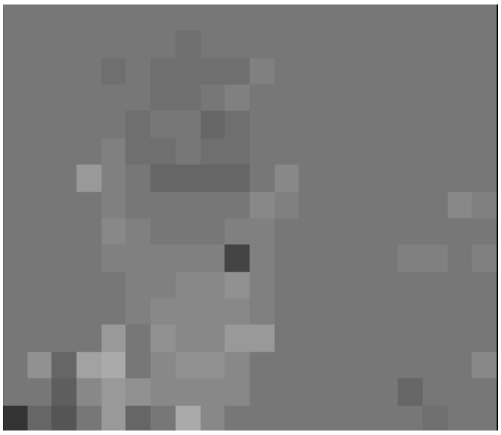
Fig. 5 Motion field of frame.17; TSBMA, 6.25 fps, window 10; (a) x-component; (b) y-component

In addition to t_m, in order to properly assess the temporal performance of the suggested algorithms it is essential to know the distribution of time values needed for determining MV for particular *MBs*, i.e. values t_i. Knowledge of this distribution is essential during real compression system design, which then has to be optimized for constant or variable flow of the output MV data. For easier interpretation the distribution of t_i values is expressed by empiric distribution function, which represents probability according to

$$P(t) = 100\ P(t_i \le t) \qquad [\%] \tag{3}$$

Here, t_i is time to determine MV of i-th *MB* and t, t_i is normalized with respect $t_{i\,max}$ (maximum t_i in a video sample).

(a)



(b)

Fig. 6 Motion field of frame.17; SSBMA, 6.25 fps, window 10; (a) x-component; (b) y-component

(a)

(b)

Fig. 7 Motion field of frame 17; PSSBMA, 6.25 fps, window 10; (a) x-component; (b) y-component

Time axis normalizing in Fig. 8 enables the comparison of particular algorithms in a single figure.

	w	*ESBMA*	*TSBMA*	*SSBMA*	*PSSBMA*
25.0 fps	10	230.8	211.1	153.9	140.0
	7	124.8	103.6	75.2	81.1
	3	31.4	23.2	16.7	20.2
12.5 fps	10	310.6	242.6	182.6	173.0
	7	122.4	118.9	89.9	89.5
	3	32.8	27.7	20.8	21.4
6.25 fps	10	310.4	202.8	182.6	177.8
	7	120.6	102.5	94.1	92.1
	3	27.8	22.3	22.0	22.7

Tab. 2 Values pf t_m [ms] for different window sizes and different frame rates

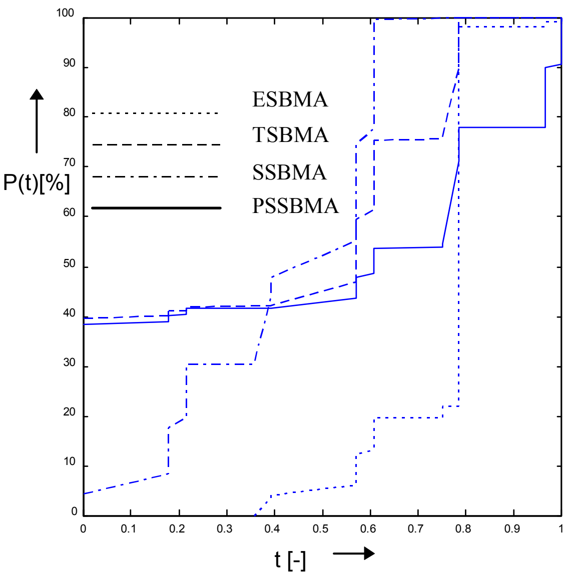


Fig. 8 Empiric distribution function

5. Discussion

TSBMA performance on Tab. 1 demonstrates the acceleration of the algorithm the tradeoff however is motion field smoothness deterioration, which is apparent comparing Fig. 5 and Fig. 4. This is caused by stopping search when meeting the MAE_{thr} criterion at search the position, which probably does not correspond to real motion.

At the same value of MAE_{thr} the SSBMA performs better than TSBMA from motion field smoothness viewpoint as shown in Fig. 6. SSBMA is also faster (Tab. 2).

Introduction of prediction by PSSBMA keeps the motion field smoothness on the same level as with SSBMA, which turns out to be better than ESBMA (Fig. 7). Prediction again accelerates the algorithm as presented in Tab. 1.

Values in Tab. 2 demonstrate that for frame rates 12.5 fps and 6.25 fps PSSBMA performs the fastest. Illustrated values demonstrate a growing trend, which is interesting especially for SSBMA and PSSBMA, where the search progresses along the search spiral. When comparing searches in two different-size search windows one would anticipate that if a certain MV is determined in the smaller window, the same MV would be determined for the larger window as well and the resulting times would be the same, as the smaller window is always contained within the larger one. This holds only if the MV is determined by meeting MAE_{thr} criterion. However, there are also *MB*s the MVs of which are determined at positions with minimum $MAE(i,j)$ values, as the criterion for MAE_{thr} can not be met within the search window. These *MB*s cause the growth of t_m at larger search windows.

Statistical behavior of the algorithms discussed is presented in Fig. 8. TSBMA and PSSBMA exhibit the larges percentage of MVs determined at the first position (for instance t=0.05). SSBMA performs much worse in from this viewpoint and with respect to the fact that PSSBMA performs the fastest it demonstrates how prediction can contribute to the overall acceleration of the MV search algorithm. PSSBMA supports thesis B.

The second important finding resulting from the comparison of the empiric distribution functions of the two fastest algorithms - SSBMA and PSSBMA, is a better support of even MV data flow achieved by PSSBMA. This is expressed by steeper development of the t_i probability density. The physical interpretation of this fact is that most MVs are found after a similar time interval. SSBMA has a less steep probability density, which implies greater variance of times to find MV. If an even flow of data from MV estimating engine is required, this SSBMA is less suitable.

6. Directions for Further Research

To enhance a prediction quality the predictor growth control mechanism should be improved. To objectively compare PSSBMA with other existing motion estimation algorithms standard video sequences should be used. To utilize achievements of the previous research, we suggest to enhance the method described in [10] and replace ESBMA by PSSBMA in order to speed up the search.

References

[1] BHASKARAN, V., KONSTANTINIDES, K. Image and Video Compression Standards. Norwell, Kluwer Academic Publishers, 1997.

[2] COTE, G., EROL, B., GALLANT, M., KOSSETINI, F. H.263+ Video Coding at Low Bit Rates. Vancouver: University of British Columbia, 1997.

[3] WIEGAND, T., LIGHSTONE, M., CAMPBELL, G., MITRA, A. A Rate-Constrained Encoding Strategy for H.263 Video Compression. In Proceedings of the International Symposium on Multimedia Communications and Video Coding. Brooklyn: Polytechnic University of Brooklyn, 1995.

[4] Overview of the MPEG-4 Standard, http://www.cselt.stet.it/ufv/leonardo/mpeg/standards/mpeg-4/mpeg-4.htm. October 1998.

[5] MPEG-4 FAQs, http://www.cselt.stet.it/ufv/leonardo/mpeg/faq/faq_mpeg-4.htm. July 1997.

[6] Video codec for audiovisual services at px64 kbps, ITU-T Recommendation H.261, 1990.

[7] Video codec for low bit rate communication, ITU-T Recommendation H.263, 1996.

[8] LI, R., ZENG, B., LIOU, M.L. A New Three-Step Search Algorithm for Block Motion Estimation. IEEE Transactions on Circuits and Systems for Video Techn. 1994, vol. 4, p. 438 - 442.

[9] PICKERING, M.R., ARNOLD, J.F., FRATER, M.R. An Adaptive Length Algorithm for Block Matching Motion Estimation. IEEE Transactions on Circuits and Systems for Video Technology. 1997, vol. 7, p. 906-912.

[10] YIH CHUAN LIN, SHEN CHUAN TAI Fast Full-Search Block-Matching Algorithm for Motion-Compensated Video Compression. IEEE Transactions on Communications. 1997, vol. 45, p. 527 – 531.

[11] MUSMANN, G., PIRSCH, P., GRALLERT, H.J. Advances in Picture Coding. Proceedings IEEE. 1985, vol. 73, p. 523-548.

[12] PRATT, W.K. Correlation Techniques for Image Registration. IEEE Transactions on Aerospace Electronics Systems. 1974, vol. 10, no. 3, p. 353-358.

[13] GHARAVI, H., MILLS, M. Block Matching Motion Estimation Algorithms – New Results. IEEE Transactions on Circuits and Systems. 1990, vol. 37, p. 649-651.

About author...

Jan PÍKA was born in 1975. From 1996 to 97 he studied at the Kansas State University in Manhattan, KS. In 1999 he graduated with honours from the Czech Technical University (CTU) in Prague; master thesis: Video Signal Compression Methods. At present he works on his PhD in the Dept. of Telecommunications at CTU in the field of video compression techniques for VLBR applications. He works as well as a Telecommunications and High-Tech consultant for Accenture.

RADIOENGINEERING REVIEWERS

April 2002, Volume 11, Number 1