True-Motion Estimation and Compensation with Multitemporal Block-Matching Search

André L. Martins
University of São Paulo
400, Trabalhador São-carlense
São Carlos - SP - Brazil

Evandro L.L. Rodrigues
University of São Paulo
400, Trabalhador São-carlense
São Carlos - SP - Brazil

Maria Stela V. Paiva
University of São Paulo
400, Trabalhador São-carlense
São Carlos - SP - Brazil

ABSTRACT
Improvements in de-interlacing algorithms are fundamental to explore image quality potential on modern TV screen technologies. This paper presents a true motion vector verification algorithm based on multi temporal block matching strategy, applied to video de-interlacing. During field scanning, a block of pixels is selected, then a block with the same reference coordinates is extracted but in the future field. A searching process runs in the previous field, looking for the most similar block; the best fit block and their coordinates are extracted from the previous field, and a new searching process is done but in the future field. If the searching process in the future field results back to the original block coordinates, the motion vector is validated; in other case, a correction over the motion vectors is done. Error calculation showed that the proposed algorithm presents image quality improvement, when compared to a classical motion compensation algorithm.

General Terms
De-interlace, motion adaptive, motion compensation, motion estimation, upscaling.

Keywords
Macroblocks, true motion vectors.

1. INTRODUCTION
TV screen technologies based on devices like plasma, LCD, or LED require video signals in a progressive scan pattern. This is due to details inherent in the technology of these screen types. Nevertheless, digital TV stations must still generate, in addition to the progressive scan standard, video transmission for interlaced field scan. International TV broadcast organizations developed standards for digital television transmission (like ATSCA/53, Part 4: 2009 [1]) including interlaced field scan as a requirement for certain image resolutions. On interlaced pattern, each video frame is split into two sets of lines, named the “fields.” For each frame, odd-numbered lines are arranged in a field called the “odd field,” and the same for even lines. To reproduce the image, CRT screens present interlaced fields following the sequence in which they are transmitted, first all odd lines and then all even lines. The interlaced scan pattern was created to minimize the visual discomfort caused by flicker of the image (which became more intense with the advent of high brightness CRT screens) without increasing the transmission bandwidth. The solution has brought major improvements at the time of their invention.

Even for video transmission format with resolution known as "Full-HD” (1920x1080), some TV broadcasting standards contemplate only in interlaced field scans. Thus, it is necessary that the TV sets (or external devices, such as set-top box) perform the conversion process from interlaced to progressive; this process is called “de-interlacing.”

De-interlacing process is not limited to rearrange line sequence before presenting frames on screen; this method, although simple, generates distortions in the image, known as “video artifacts.” Many distortions generated by the process of de-interlacing are caused by moving objects in the image. Several algorithms have been developed in an attempt to correct these distortions, but there is still much to be explored.

In many cases, it is necessary to apply processes similar to those used in the algorithms of de-interlacing on the appropriateness of video formats for presentation at higher-resolution screens, with the smaller generation of artifacts as possible. Such processing is known as “video upscaling.” By its complexity, certain video-upscaling algorithms are implemented in hardware (at least partially), as described by Ramadevi [2].

During the processes of de-interlacing and upscaling, it is necessary to estimate missing lines in the fields of interlaced scanning. It is an attempt to reconstruct the image details that were not transmitted due to limitation of the bandwidth, or who have not even been captured by the camera. According to Bellers [3], the process of de-interlacing "...doubles the vertical sampling density."

Advanced algorithms of de-interlacing are able to identify regions of the image where there is movement, and apply the so-called "motion compensation"; such algorithms require the determination of true-motion vectors. In this paper, the development of a verification algorithm of motion vectors is presented by the strategy known as “block-matching search” that identifies the movement of structures (macroblocks) equivalent between the fields. The presented method is based on motion-estimation algorithm proposed by De Haan [4], and contributes to it by verifying the vectors by searching the block (found in the previous field) in the posterior field, after the searching process in the previous field, used for determining the block of greater compatibility.

In the next section, a classification and a basic description of de-interlacing methods are presented that are used as the basis for the development of the proposed algorithm.

2. DE-INTERLACING METHODS
2.1 Simple De-Interlacing
In simple de-interlacing method, basically ordering of lines occurs; odd field lines (1, 3, 5, ...) are interspersed with the even field lines (2, 4, 6, ...), forming a complete progressive frame (with the lines in numerical sequence (1, 2, 3, 4, 5, 6, ...). When this frame is presented on a screen of plasma, LCD, or LED, video artifacts appear in regions where there is motion from one field to another. These flaws arise from the fact that the two fields that have generated the frame were acquired in different moments, and this temporal difference is greater than it would be if the image was captured in progressive scan and transmitted with double bandwidth. In

addition to the temporal difference of the capture of the fields, the generated sequence of progressive frame lines does not match the temporal order in which they were captured by the camera.

There is therefore a temporal mismatch between the fields, and if there is motion of objects (or global motion due to camera movement) between the moments of acquisition of two fields, the reorganization of lines generates a double image effect, or "feathering effect" on parts of objects or people who move from one field to another. This effect is worse when the speed of the objects is higher, and is barely noticeable on screens of type CRT, by the technology itself, and also by the fact that the lower resolution of this type of screen does not allow a direct comparison of image quality at all resolutions available in other more modern screen technologies.

2.2 De-Interlacing by Intrafield Median (Spatial Median) or Interfield Median (Temporal Median)

For intrafield median, the nonexistent lines are reconstructed on the basis of the neighboring lines on the same field (spatial neighborhood), or even between neighboring lines on previous or later fields (temporal neighborhood). In both cases, blurring occurs on the image, because this type of procedure produces the effects of a low-pass filter.

The image artifacts generated can be worse than those created by simple de-interlacing, because the blurring occurs only in half of the lines, since the other half consists of original lines of the field.

2.3 De-Interlacing by Motion Estimation

The concept of motion estimation was initially developed for identifying patterns of repetition in video, as described by Mounts [5], and for estimation of displacement from one frame to another, as described by Netravali [6], but with the final goal of reducing TV broadcasting bandwidth. The basic idea is to transmit only information pertaining to areas of the image where there is movement, and the rest of the image is constructed simply by repeating the pixels of the previous field.

The use of motion estimation to handle video signals emerged later, having as featured Netravali work [7], which developed and patented a method to perform motion estimation by linear interpolation between consecutive fields.

There was a leap in evolution in the algorithms of video upscaling and de-interlacing since the emergence of true-motion estimation technology, designed to reduce errors in the estimation of motion vectors. Many motion-vector algorithms are based on the search for similar blocks (block-matching search) proposed by Vandendorpe [8]. In this type of algorithm, the current field is divided into blocks of pixels; each block is selected and the fields previous and/or posterior are swept away (within a predefined neighborhood) in search of a block with maximum similarity.

The modern processes of upscaling and de-interlacing require motion-estimation algorithms as a fundamental need. Such algorithms use motion estimation to nonexistent lines production in the original video signal that complements each video frame.

2.4 Motion-Adaptive Algorithms and Motion Compensation

The concept of motion-adaptive algorithms refers to processes that identify areas of the image that have movement and perform a spatial interpolation in these regions. In regions without movement, temporal interpolation is made. These algorithms tend to blur the image in moving regions.

Already, the so-called motion-compensation algorithms temporally interpolate the areas with movement using motion vectors. For this to be possible, it is necessary to determine true-motion vectors, to identify the right coordinates of a moving object in each field. The process of interpolation is more controlled, and should consider the changes of coordinates of the block equivalent of a field to another due to the motion detected.

There are several cases in which the development of de-interlacing algorithms based on motion estimation and motion compensation resulted in the creation of dedicated systems on a chip, with the objective of increasing its performance. One of these developments is described by De Haan [9].

In the next section, the details of the proposed algorithm and the strategy used for performance verification are presented.

3. TRUE MOTION VECTOR VERIFICATION

3.1 Base Algorithm

The method of motion compensation proposed by De Haan [4] was used as a base by several researchers. In this method, each video field is subdivided into macroblocks, of a size that ensures that the objects in the image are larger than the size of the macroblock. In Fig. 1, a hypothetical field is presented, which was divided into macroblocks of arbitrary size.

Field n

$B_{1,1}$	$B_{1,2}$	$B_{1,3}$	$B_{1,4}$	$B_{1,5}$	$B_{1,6}$	$B_{1,7}$	$B_{1,8}$	$B_{1,9}$	$B_{1,10}$	$B_{1,11}$	$B_{1,12}$	$B_{1,13}$	$B_{1,14}$	$B_{1,15}$	$B_{1,16}$
$B_{2,1}$	$B_{2,2}$	$B_{2,3}$	$B_{2,4}$	$B_{2,5}$	$B_{2,6}$	$B_{2,7}$	$B_{2,8}$	$B_{2,9}$	$B_{2,10}$	$B_{2,11}$	$B_{2,12}$	$B_{2,13}$	$B_{2,14}$	$B_{2,15}$	$B_{2,16}$
$B_{3,1}$	$B_{3,2}$	$B_{3,3}$	$B_{3,4}$	$B_{3,5}$	$B_{3,6}$	$B_{3,7}$	$B_{3,8}$	$B_{3,9}$	$B_{3,10}$	$B_{3,11}$	$B_{3,12}$	$B_{3,13}$	$B_{3,14}$	$B_{3,15}$	$B_{3,16}$
$B_{4,1}$	$B_{4,2}$	$B_{4,3}$	$B_{4,4}$	$B_{4,5}$	$B_{4,6}$	$B_{4,7}$	$B_{4,8}$	$B_{4,9}$	$B_{4,10}$	$B_{4,11}$	$B_{4,12}$	$B_{4,13}$	$B_{4,14}$	$B_{4,15}$	$B_{4,16}$
$B_{5,1}$	$B_{5,2}$	$B_{5,3}$	$B_{5,4}$	$B_{5,5}$	$B_{5,6}$	$B_{5,7}$	$B_{5,8}$	$B_{5,9}$	$B_{5,10}$	$B_{5,11}$	$B_{5,12}$	$B_{5,13}$	$B_{5,14}$	$B_{5,15}$	$B_{5,16}$
$B_{6,1}$	$B_{6,2}$	$B_{6,3}$	$B_{6,4}$	$B_{6,5}$	$B_{6,6}$	$B_{6,7}$	$B_{6,8}$	$B_{6,9}$	$B_{6,10}$	$B_{6,11}$	$B_{6,12}$	$B_{6,13}$	$B_{6,14}$	$B_{6,15}$	$B_{6,16}$
$B_{7,1}$	$B_{7,2}$	$B_{7,3}$	$B_{7,4}$	$B_{7,5}$	$B_{7,6}$	$B_{7,7}$	$B_{7,8}$	$B_{7,9}$	$B_{7,10}$	$B_{7,11}$	$B_{7,12}$	$B_{7,13}$	$B_{7,14}$	$B_{7,15}$	$B_{7,16}$
$B_{8,1}$	$B_{8,2}$	$B_{8,3}$	$B_{8,4}$	$B_{8,5}$	$B_{8,6}$	$B_{8,7}$	$B_{8,8}$	$B_{8,9}$	$B_{8,10}$	$B_{8,11}$	$B_{8,12}$	$B_{8,13}$	$B_{8,14}$	$B_{8,15}$	$B_{8,16}$
$B_{9,1}$	$B_{9,2}$	$B_{9,3}$	$B_{9,4}$	$B_{9,5}$	$B_{9,6}$	$B_{9,7}$	$B_{9,8}$	$B_{9,9}$	$B_{9,10}$	$B_{9,11}$	$B_{9,12}$	$B_{9,13}$	$B_{9,14}$	$B_{9,15}$	$B_{9,16}$

Figure 1: A hypothetical field divided into macroblocks.

Assuming that the current field is odd, the even-numbered lines have to be estimated in some way, so that it is mounted as the complete frame. To do this, the field is scanned, and a block-by-block searching process is done, looking for the most compatible block in the previous field (n-1).

Two fields are needed to estimate the missing lines (in addition to the present field being scanned). The procedure is done as follows: block scanning is made in the current field n; block coordinates (x, y) are selected, and the value of the pixels of the same coordinates is taken but in the posterior field (n +1), as shown in Fig. 2.

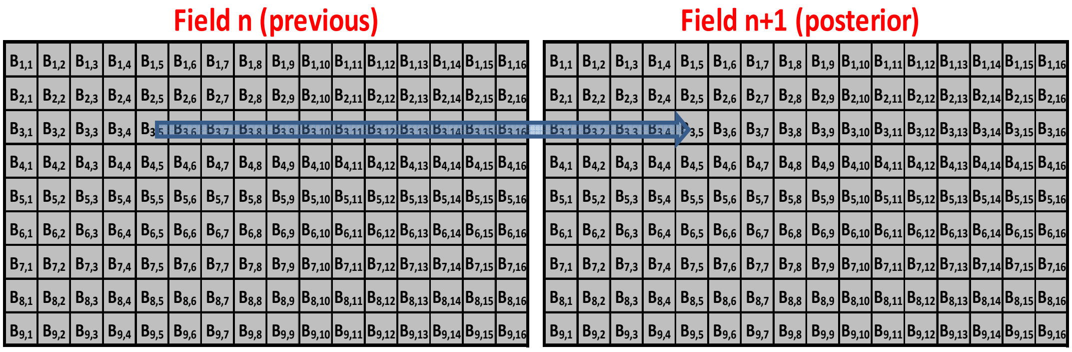

Figure 2: Extracting value from pixels of a block in the posterior field (n+1).

Once extracted, the value of block pixel coordinates (x, y) in the posterior field (n + 1) returns to the previous field (n-1), defines a search neighborhood (in pixels), and the search is performed by the block with greater resemblance, as shown in Fig. 3.

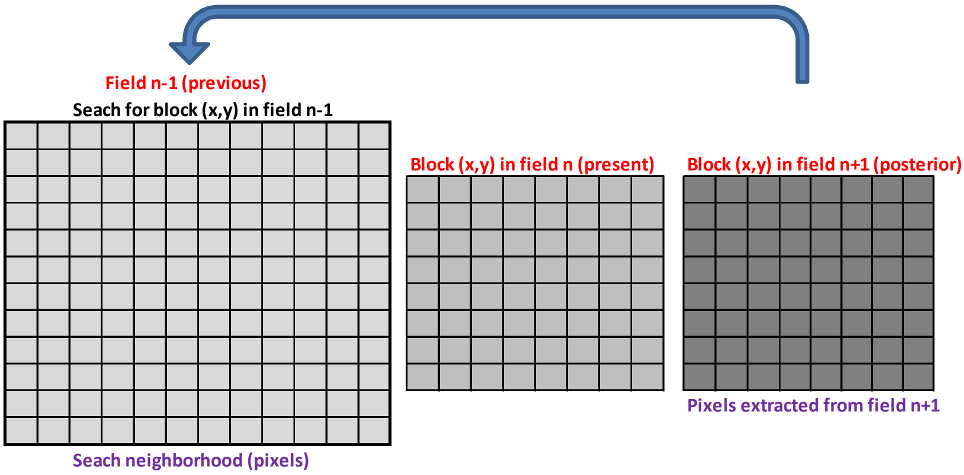

Figure 3: Block search (x,y) in the previous field (n-1).

The searching neighborhood involves a number of candidates, what defines the "candidate set." There are various methods of searching, each having different strategies to minimize the processing time. As described by Bellers [3], the exhaustive search strategy was used, and the mean square error (MSE) was used as a criterion for matching. The calculation of the MSE is presented in (1).

$$MSE = \sum_{\vec{x} \in B(\vec{X})} (F(\vec{x}, n+1) - F(\vec{x} - \vec{C}, n-1))^2 \quad (1)$$

Where:

- *MSE = mean square error between the block in the field n + 1 and the candidate block in the field n-1;*
- $\vec{x}$ *= pixel coordinates in the field n;*
- *B = selected block in the field n;*
- $\vec{X}$ *= block coordinates in the field n;*
- *F = pixel level;*
- *n = current field number;*
- $\vec{C}$ *= motion vector of candidate block in the field n-1;*

At the end of the search, we identify the equivalent block as the one that presents the smallest MSE, and use the coordinates of this block as the motion vector of all pixels in the block of coordinates (x,y) on the current scan. If the motion vector points to the same coordinates as that of the original block, it is considered that there was no movement, and the missing pixels in the field n are estimated by the repetition of the pixels in the field n-1.

As indicated by Bellers [3], when motion is identified, the process of motion compensation by means of estimation of pixels on missing lines by the median between the two neighboring lines of current field n (spatial neighborhood) and the corresponding line in the previous field (n-1) is identified by the coordinates of the motion vector. This calculation is presented in (2).

$$F_i(\vec{x}, n) = MED\{F(\vec{x} - \vec{u}_y, n), F(\vec{x} + \vec{u}_y, n), F(\vec{x} - \vec{C}, n-1)\} \quad (2)$$

Where:

- F_i *= pixel estimated level;*
- $\vec{x}$ *= pixel coordinates;*
- *n = current field number;*
- *MED = median calculation;*
- *F = pixel level with coordinates* $\vec{x}$*;*
- $\pm\vec{u}_y$ *= vector that points to pixel in the same column as pixel* $\vec{x}$*, but on line down/up;*
- $\vec{C}$ *=motion vector of candidate block in the field n-1;*

This estimation process is represented in Fig. 4.

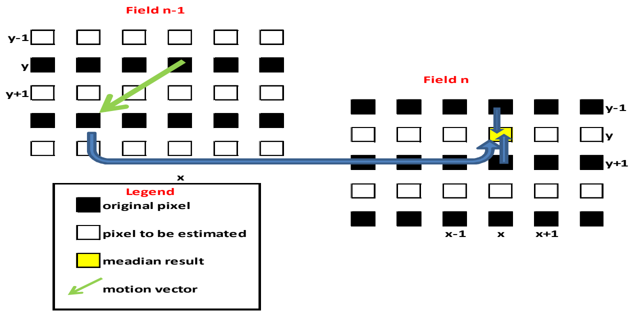

Figure 4: Estimation of a pixel by the median of the neighborhood, considering the true-motion vector.

3.2 The proposed algorithm

In the proposed algorithm, a similar block found in the previous field (n-1) is extracted, usually as performed in the method of de Haan [4], and then a new search is carried out, but now in the posterior field (n + 1), from which the block sought in the previous field was extracted. This new search aims to confirm the motion vector by checking if the vector of new search points to the original position of the block in the posterior field (x, y). This process is shown in Fig. 5.

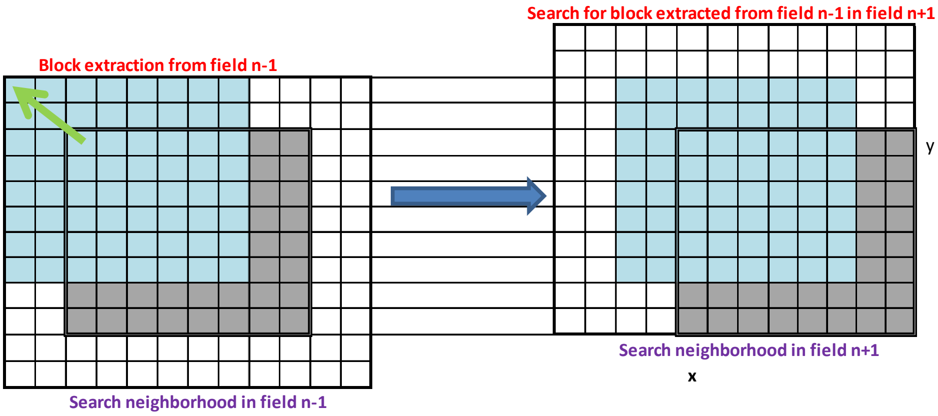

Figura 5: Search for the block extracted from the previous field (n-1) in the posterior field (n + 1) for confirmation of motion vector.

On several occasions, it can happen that the block extracted in the previous field (n-1) has greater similarity with the blocks other than the block of source coordinates (x, y). If the coordinates of the block found in this new search have the same source block coordinates, the motion vector is considered consistent, being then validated. The estimation of missing lines of the current frame is then accomplished by calculating the median of neighboring lines spatially and temporally. In the situation where the search for the field block n + 1 results in different coordinates of the source block, partial invasion occurs in other macroblocks, as shown in Fig. 6.

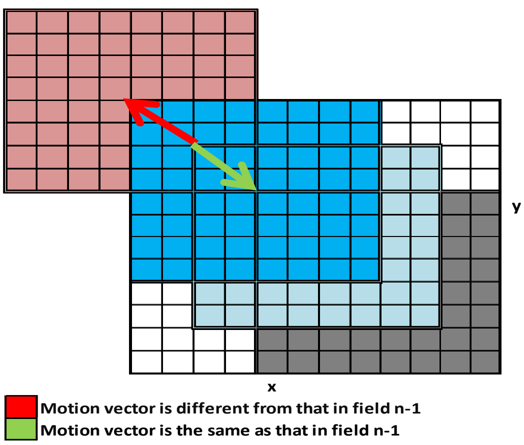

Figure 6: Search for block on return to the field n + 1 can result in different vectors of the original; the block found partially invades other blocks.

If this situation of inconsistency occurs, the original motion vector is not confirmed, and the pixels of a similar block search in returning to the posterior field have their motion vector corrected. In this way, the estimation of missing lines of the current frame is performed by the same method of the median, but considering the corrected motion vector.

This process occurs block by block, until the entire video field is swept and all the missing lines are estimated.

3.3 Performance Evaluation of the Proposed Algorithm

To calculate the estimated line errors and make comparison between the algorithms, we created artificial interlaced fields from video files. The procedure adopted was as follows: select a video file and remove the odd-numbered lines of the even frames and even-numbered lines of odd frames, producing the artificial fields. These fields were subjected to motion estimation and de-interlacing algorithms.

The frames generated by the algorithms were compared with the original video frames, determining the error by the sum of absolute differences (SAD). The calculation of SAD between the original frames and the estimated frames is presented in (3).

$$SAD = \sum \left| F(\vec{x},n) - Fi(\vec{x},n) \right| \qquad (3)$$

Where:

- *SAD = sum of absolute differences;*
- *F= pixel level on original frame;*

- $\vec{x}$ = *pixel coordinates;*
- *n = current frame number;*
- *Fi= pixel level on estimated frame;*

The proposed algorithm has been implemented in the MatLab platform [10], and its performance was compared with the following methods: simple de-interlacing, spatial median, temporal median, and motion-compensation algorithm from De Haan [4]. The macroblock size has been set as a parameter, as well as the search neighborhood dimension. The algorithm also allows one to define neighborhoods of different sizes in the previous field search and in the search in posterior field (confirmation). The errors were compared in various combinations of blocksize, size of the neighborhood of previous field search, and search of neighborhood size in the posterior field.

Artificial interlaced frames were primarily generated for all videos in that the proposed algorithm has been applied. As an illustration of the process, we present one of the frames used during the development; the file is "viplane.avi," native file from Matlab platform; this is an 80 frame video, with 168 lines and 360 columns. Fig. 7 shows a frame of the original video.

Figure 7: Original frame of the video "viplane.avi", native from MatLab platform.

The video is processed and converted, so that each original frame generates a single artificial field. Only the odd-numbered lines of the odd frame are used, forming an artificial odd field with half the original lines. The same happens with the even frames, in which case even lines are extracted, forming the even artificial field. Fig. 8 presents a frame generated by simple de-interlacing method.

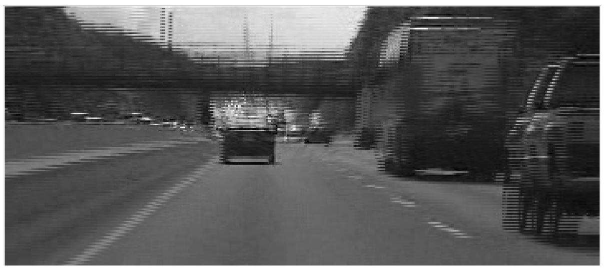

Figure 8: Frame produced by the artificial fields.

It is possible to notice the video artifacts generated by simple de-interlacing, like double image and "feathering effect," in regions of objects in motion.

Once the artificial fields are generated, the de-interlacing algorithm is applied; first motion-estimation algorithms and generation of vectors are performed; from the motion vectors obtained, the de-interlacing algorithm is applied. As the information of the original lines is known (disposed to generate artificial fields), it is possible to determine the exact error of missing lines, which were generated by the de-interlacing algorithm.

In the next section, we present the results of the performed tests and the comparison with the base algorithm from De Haan [4].

4. TEST RESULTS

Among the various tests performed, we present here the results of applying the proposed algorithm in five video files. Initially, the algorithm was set to a macroblock size of 8 x 8, as indicated by De Haan [4]; during development, the macroblock size was varied from 3x3 to 16x16, and the neighborhood ranged from 1 to 5 pixels larger than the macroblock size. These parameter changes had a great influence on the processing time, but little influence on the results in terms of image quality, which encourages better further investigation. The low sensitivity of the algorithm to the size of the macroblocks, since they are smaller than the objects in the image, was evidenced by De Haan [4]. Fig. 9 shows a frame generated by the proposed algorithm, applied to the video "viplane.avi."

Figure 9: A frame generated by the proposed algorithm.

It can be observed in Fig. 9 that the proposed algorithm was able to fix several video artifacts, though retaining regions with the “feathering effect.”

For identification of regions of the image where the proposed algorithm operates effectively, an extra video frame was generated indicating the macroblocks where "no confirmation" of the original vectors occurred.

Fig. 10 shows the regions of the image in which the proposed algorithm held correction of motion vectors.

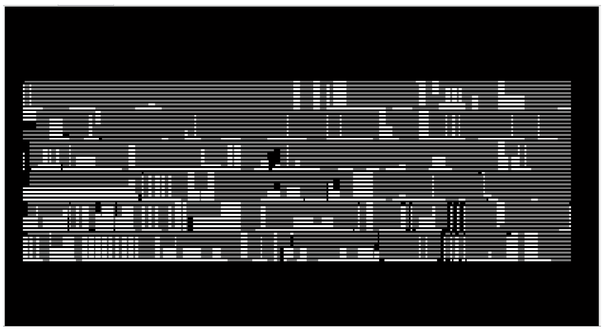

Figure 10: Regions of the image where there is correction in motion vectors.

In addition to the method from De Haan [4] and the proposed method, we also implemented the methods of simple de-interlacing, estimation method by spatial median, and method for temporal median (all on Matlab platform).

To measure the performance and compare the errors of estimation of lines between several algorithms, we carried out the calculation of SAD, as (3), for each frame estimated in relation to the frame of the original video. We created a variable that accumulates this frame-by-frame value so that at the end of the processing of the full video, this variable will contain the value added from differences of all frames. Table I shows the accumulated SAD when compared with the original video, for five different videos submitted to five methods of de-interlacing implemented.

Table I Comparative error of de-interlacing methods implemented

Acumulated Sum of Absolute Differences						
Video file	Interlaced	Spatial median	Temporal median	MC Haan	Proposed MC	% MC's
viplane.avi	70.685	65.800	57.684	44.855	43.836	-2,27
viptrain.avi	492.600	1.203.100	380.680	393.290	391.720	-0,40
viplanedeparture.avi	243.890	234.170	180.890	136.820	130.720	-4,46
viptraffic.avi	35.916	54.118	29.855	21.285	20.591	-3,26
vipwarnings.avi	479.700	472.490	396.510	314.090	308.730	-1,71

The right column of Table I presents the percentage error of the algorithm proposed in relation to the algorithm of De Haan [4].

Another implementation also performed and tested refers to scan the field and selection of macroblocks. Instead of scanning the field with an offset equal to the dimension of the macroblock, the scan was done with offsets equal to half the size of the macroblock. This strategy did not provide significant decrease in errors, besides having the disadvantage of increasing the processing time.

5. CONCLUSIONS

It has been shown that the proposed verification algorithm of motion vectors featured image quality gains in relation to the method from De Haan [4], contributing to the advancement of methods of video de-interlacing and upscaling.

Other strategies may be incorporated into the proposed method to reduce errors of motion vectors, reducing the processing time of the search algorithm, and de-interlacing algorithm, as proposed by Chen [11]. Chen presents methods to identify regions of the image that, even after the application of the algorithm of motion compensation, will feature defects visible, as the "feathering effect."

In future work, an analysis of processing time compared with other methods will be carried out. Also, tests shall be done with the inclusion of iterative corrections, performing several searches of the macroblock in the anterior and posterior fields, until the convergence of motion vectors. It is also necessary to include protections against errors of estimation of motion vectors, as quoted by Bellers [3].

The influence of the size of the search window in image quality, both in the previous and posterior field, also requires further investigation.

In future work it is important to investigate the performance of the algorithm in videos with higher definition, and other occlusion situations.

It is intended to combine the advantages of the proposed algorithm with other de-interlacing strategies, so as to obtain enough image quality improvement to be worth to include it in video signal processors for TV sets.

6. REFERENCES

[1] Advanced Television Systems Committee. About standard ATSC A/53, Part 4:2009, "MPEG-2 Video System Characteristics". ATSC Website. http://www.atsc.org/cms/standards/a53/a_53-Part-4-2009.pdf (2014).

[2] Ramadevi, G. and T.Mallikarjuna, T. 2012. Real time artifact-free image upscaling. Journal of Electronics and Communication Engineering, vol. 3, Issue 1, pp.12-19.

[3] Bellers, E.B. and De Haan, G. 2000. De-interlacing: A key technology for scan rate conversion," Elsevier, Amsterdam.

[4] De Haan, G., Biezen, P. W. A. C., Huijgen, H. and Ojo, O. A. 1993. Truemotion estimation with 3-D recursive search block matching. IEEE Trans. Circuits Syst. Video Technol., vol. 3, no. 5, pp. 368–379.

[5] Mounts, F.W. 1969. A video encoding system with conditional picture-element replenishment. The Bell System Technical Journal, 48, pp. 2545-2554.

[6] Netravali, A.N. and Robbins, J. D. 1978. Motion compensated television coding: Part I. The Bell System Technical Journal, 58, pp. 631-670.

[7] Netravali, A. N. et al. 1983. Video signal interpolation using motion estimation. US Patent 4,383,272.

[8] Vandendorpe, L. et al. 1994. Motion-compensated conversion from interlaced to progressive formats. Signal Process.: Image Commun., vol. 6, pp. 193-211.

[9] De Haan, G., Kettenis, J., Loning, A. and De Loore, B. 1996. IC for motion-compensated 100Hz TV with natural-motion movie-mode. IEEE Transactions on Consumer Electronics, vol. 42, pp. 165-174.

[10] The MathWorks, Inc. About MatLab platform. Mathworks web site. http://www.mathworks.com/products/matlab/. (2014).

[11] Chen, Y.-R. and Tai, S.-C. 2009. True motion compensated de interlacing algorithm. IEEE Trans. On Cir. & Syst. For Video Tech, vol. 19, no. 10, pp. 1489-1498.