Temporal Interpolation In Mobile and Internet Video Services

Volodymyr Moroz, *Member, IEEE*

Abstract — **The inter-frame prediction modified algorithm with the motion compensation based on the multi-resolution wavelet analysis is proposed. Segmentation and optical flow tracking is used for predicting of the inter-frame image. For the motion compensated prediction the algorithm roughly matches the wavelet low scale domain blocks and then the matching is doing more precisely by extract the additional information about motion objects in the higher scale wavelet domains. It helps to increase the prediction quality. Furthermore, to compensate the edge artifacts the non-homogeneous double cross search of the spatial-temporal frame information is used.**

Keywords — **Temporal interpolation, motion compensation, wavelet transformation, multi-resolution analysis.**

I. Introduction

One of the challenging problems that majority up-to-data video processing technologies use is the frame rate up-conversion, which converts the video frame rate from a lower number into a higher one [1]. This problem arises in the low bit-rate video transmission systems, where the frame rate at the encoder is down-converted by skipping frames to achieve lower transmitted bandwidth, and afterward the frame interpolation is used in the decoder to restore the skipped frame. Another application is the video-data conversion from one TV standard to another (for example, PAL into NTSC).

To restore the skipped frame, the interpolation problem should be solved in the decoder, where the appropriate time series is a brightness profile that changes in time from one frame to another. However, during the interpolation procedure, some undesirable distortions are accomplished the recovering video, such as, object overlapping, edge artifacts, motion blur and so on. Hence, the quality of the restored video is not always acceptable.

In practice, instead of the classical interpolation, the motion compensation interpolation is used. The motion compensation interpolation is based on the image analysis, that is, detecting of the moving objects, and then tracking them by estimation of their velocity from frame to frame [2]. Assuming that the object motion can be expressed in terms of a velocity vector in the spatial-temporal domain, the velocity estimation problem leads to the measurement of the velocity vector magnitude and its direction. As a rule, for the sake of convenience, the vertical and horizontal velocity vector increment between two adjacent frames is estimated. The resulting vertical and horizontal coordinate increments are the set of the velocity vectors in the Cartesian coordinates.

V. Moroz is with the Computational Mathematics Department, Odessa National University, 2, Dvoryanska str., Odessa, 65026, Ukraine; (phone: 38-067-4846975; fax: 38-048-2378646; e-mail: v.moroz@onu.edu.ua).

The simplest video scenario is where all motion objects are moved in the same direction with the same velocity; but, in general case, the objects move randomly. Therefore, the image is split in the set of K nonintersecting motion regions (blocks) in which either the object or set of objects have the same velocity vector. So, the object or the set of the objects can be represented by their velocity vector inside the independent non-overlapping blocks. The greater number of the blocks with corresponding velocity vectors the better compensation of the motion effect can be achieved. The boundary case is when the number of independent velocity vectors is equal to the number of the image pixels. If the velocity vectors for all image pixels are known then the inter-frame image can be predicted precisely by moving those vectors to the required points.

The faithful estimation of the velocity vectors can be found as a projection of the optic flow equation to the multi-resolution wavelets. In [3], [4] the fast algorithm that compute the optic flow in the wavelet segments is proposed. The main drawback of such kind of algorithms is that the small objects as well as the fast moving objects can be lost when the wavelet filtration with the large-scale blocks is used.

In this paper we propose the modified inter-frame motion-compensated prediction algorithm that uses the multi-scale wavelet analysis and based on the matching procedure over the wavelet transform of the low frequency domain segments. The main point of the modification is the exploiting of the additional information about motion objects inside the wavelet higher frequency segments by finding the projection of the optical flow to the multi-resolution wavelets, where the scale is reduced gently, and as a consequence, the motion can be estimated more precisely. By doing so, we achieve some additional goals in the block matching procedure. Afterward, to increase the prediction quality we use the non-homogeneous double cross search of the spatial-temporal frame information in the wavelet high frequency segments of the multi-scale wavelet analysis.

II. Block Matching Problem

Let us consider an optical flow as a three dimensional function $f(\mathbf{x}, t)$, where $\mathbf{x} = (x_1, x_2)$ is a spatial two dimensional vector of a pixel coordinates, and t is a time coordinate, i.e., the function $f(\mathbf{x}, t)$ can be considered as an image density or brightness profile in the spatial-temporal coordinate. Forming of video stream requires the optical flow sampling (framing) in both the space and the time domain. The video stream can be considered as a set of a moving from frame to frame pixels, each with corresponding trajectory between two adjacent frames; it forms the set of optical flow velocity vectors [5].

With known optical flow vectors each pixel can be precisely tracked from one frame to another by following the velocity vector; hence, quite accurate prediction can be done for the majority pixels within the inter-space of the adjacent frames. However, enormous computational complexity is expected when the algorithm tries to track all valuable pixels of the optical flow. Therefore, in practice, instead of the solitary pixel processing the block of the pixels as a motion unit is used [2]. The block size is chosen in order to all pixels inside the block have the same degree of motion, and, as a result the same velocity vectors. Various approaches exist in the literature to compute the velocity vector inside the block, including the estimation of the total increment as well as differential increment of the optical flow [2], [5], [6].

To obtain the precise inter-frame prediction with the motion compensation the block matching problem should be solved. For each block in a frame, a search is made in the reference frame for the best matching block, minimizing least mean square prediction error. A good match means that the block has undergone the same translation, and in the same time the block is not overlap the objects that have different degrees of motion.

Firstly we discuss the block matching problem with minimize the least mean square difference prediction error for some fixed block size [2]. Let $f_k(\mathbf{x}, i\Delta)$ be a brightness profile of the k-th motion region, k = 1, 2, …, K inside the i-th frame and Δ be a frame interval. Suppose that all pixels of the k-th motion region have the common velocity vectors. If during the time range, Δ, the brightness offset vector $\mathbf{m}_k(\mathbf{x}) = (m_1, m_2)$ is nonzero (or acceded some threshold) then the appropriate brightness profile of the k-th motion region at the (i+1)-th frame is $f_k(\mathbf{x}, (i+1)\Delta) = f_k(\mathbf{x} + \mathbf{m}_k(\mathbf{x}), i\Delta)$, and corresponding velocity vector for k-th motion region can be found as $\mathbf{v}_k(\mathbf{x}, \Delta) = \mathbf{m}_k(\mathbf{x})/\Delta$.

Because of insignificant changing of the scene during the time range Δ let us assume that the velocity vector changes inside some small vicinity $\boldsymbol{\delta} = (\delta_1, \delta_2)$, $\delta_1, \delta_2 \in \mathbf{z}$, and $\forall \mathbf{x} \in f_k(\mathbf{x}, \Delta i)$, $|\mathbf{x} - \boldsymbol{\delta}| \in \mathbf{z}$, where $\mathbf{z}$ is enough small region motion. Then the corresponding offset vector is $\mathbf{m}_k(\boldsymbol{\delta}) = (m_1, m_2)$, where $m_1, m_2 \in \mathbf{z}$, and velocity vector for k-th motion region is $\mathbf{v}_k(\boldsymbol{\delta}, \Delta) = \mathbf{m}_k(\boldsymbol{\delta})/\Delta$. Therefore, the brightness profile of the k-th motion region for the (i+1)-th frame can be approximated as $f_k(\mathbf{x}, (i+1)\Delta) \approx f_k(\mathbf{x} + \mathbf{m}_k(\boldsymbol{\delta}), i\Delta)$. Generally the motion regions is approximated by square blocks with size z, so $\mathbf{z} = (z, z)$.

Hence, the block matching problem leads to minimization of the differential norm between motion regions brightness profiles for the adjacent frames at the instants of time $(i\Delta)$ and $(i+1)\Delta$, i.e.,

$$\varepsilon(\boldsymbol{\delta}, \mathbf{m}(\boldsymbol{\delta})) = \iint\limits_{\substack{|\mathbf{x}-\boldsymbol{\delta}| \le \mathbf{z} \\ k=1,2,\dots,K}} \left| f_k(x_1, x_2, i\Delta) - f_m(x_1 + \delta_1, x_2 + \delta_2, (i+1)\Delta) \right|^2 dx_1 dx_2 , \quad (1)$$

where δ_1, δ_2 is approximated by staircase function.

As was mentioned, the scene is not changing significantly during the time Δ, hence in (1) only the nearest motion regions inside some small vicinity $\delta_1, \delta_2 \le z$, can be considered. Finally, minimizing the error in (1) along all possible offset vectors,

$$\varepsilon(\boldsymbol{\delta}, \mathbf{m}_k(\boldsymbol{\delta})) = \arg \min_{\forall\, \mathbf{m}(\delta)} \varepsilon(\boldsymbol{\delta}, \mathbf{m}(\boldsymbol{\delta})), \qquad (2)$$

leads to the estimation of the required offset vectors, $\mathbf{m}_k(\boldsymbol{\delta})$, that solves the block matching problem.

Solution of the block matching problem requires choosing the matching region z, which provides the information for a motion compensation algorithm. Choosing of the size z is a quite challenging problem because it determines the maximum value of the velocity vector, since it bounds the maximum velocity of the moving objects. Furthermore, the size z determines both the interpolation quality and computational complexity of the processing algorithm. On the one hand, when z is chosen very small, the error function $\varepsilon(\boldsymbol{\delta}, \mathbf{m}(\boldsymbol{\delta}))$ has several local minimum, because of a large number of very likely blocks. These minima quite difficult to discriminate due to both very close their amplitudes and the noise influence. It can lead to erroneous estimation of the velocity vector, $\mathbf{v}_k(\boldsymbol{\delta}, \Delta)$. On the other hand, when z is chosen very large, the velocity can be found more precisely, but the resolution of the resulting video for many applications is unacceptable. For the fixed size z the computational complexity of the compensated motion algorithm is $O(z^2 N^4)$, where N is frame size in units of pixels [3]. So, it is enormously large. Some computational saving is proposed in [6], [7], [8].

To obtain further computational saving of the motion compensated algorithm that based on the block matching the multi-resolution image approximation (MIA) can be used [3], [4], [9]. Let us consider the brightness profiles of two matched blocks within the wavelet scale 2^j, i.e., $f_k^j(\mathbf{x}, i\Delta)$ and $f_k^j(\mathbf{x}, (i+1)\Delta)$. MIA algorithms primarily compute the rough approximation of the velocity vector of the j scale. For that, $f_k^j(\mathbf{x}, i\Delta)$ and $f_k^j(\mathbf{x}, (i+1)\Delta)$ is roughly matched by mean of square error minimization inside the blocks of size z. Afterwards, the blocks are matched more precisely within the scales $2^{j-1}, 2^{j-2}$.

Hence, MIA starts to match the larger size blocks, and then block size diminished when the scale step-down. The required computational work of such sort of algorithms is $O(K^2 z^2 N^2)$ [10], where K^2 number of nonzero (which exceeded the threshold) is offset vectors $\mathbf{m}(\delta)$ and which satisfy the matching requirements (2). The main drawback of such kind of algorithms is that the small objects as well as the fast moving objects can be lost inside the large-scale wavelet blocks that leads to video image distortion, some time significantly.

III. Wavelet Motion Estimation

Taking into account that frame brightness profile $f(\mathbf{x},t)$ is smooth in time function and almost unchangeable inside the couple of frames, the total increment can written as

$$\frac{df(\mathbf{x},t)}{dt}=\frac{\partial f(\mathbf{x},t)}{\partial x_1}x_1'(t)+\frac{\partial f(\mathbf{x},t)}{\partial x_2}x_2'(t)+\frac{\partial f(\mathbf{x},t)}{\partial t}=0\,. \quad (3)$$

Then, the optical flow expression that joint together the time *t* with both the velocity vector and brightness level in any arbitrary frame image point becomes

$$\vec{\nabla} f\cdot\mathbf{v}=\frac{\partial f(\mathbf{x},t)}{\partial x_1}v_1+\frac{\partial f(\mathbf{x},t)}{\partial x_2}v_2=-\frac{\partial f(\mathbf{x},t)}{\partial t}\,,$$

(4)

where $\mathbf{v}=(v_1,v_2)=(x_1',x_2')$ is a velocity vector, $\vec{\nabla} f=\left(\frac{\partial f}{\partial x_1},\frac{\partial f}{\partial x_2}\right)$ is a gradient of f .

If brightness profile $f(\mathbf{x},t)$ is smooth in time function then $\mathbf{v}$ is a smooth vector-function of **x**, and it can be approximated by staircase function with the small enough step. The accurate estimate of the velocity vector can be found as a projection of the optic flow equation to the multi-resolution wavelets [3], [4], [11].

Let us consider the mother wavelet

$$\psi_{\boldsymbol{\delta},\mathbf{z}}=\frac{1}{z}\psi\left(\frac{x_1-\delta_1}{z},\frac{x_2-\delta_2}{z}\right). \quad (5)$$

We assume that the shift of the processing image during the time Δ is small enough related to the wavelet scaling size, which is *z*. The size *z* is determined with suggestion that the matching mean square error is small enough when $\left|\mathbf{v}(\boldsymbol{\delta},t)\right|\le \mathbf{z}/\Delta$, and the value $\mathbf{v}(\boldsymbol{\delta},t)$ is almost unchangeable inside the scale of $\psi_{\boldsymbol{\delta},\mathbf{z}}$. Computing the inner product of the optical flow (4) and the wavelet (5), and afterward using partial integration we get

$$\left\langle f(\mathbf{x},t),\frac{\partial\psi_{\boldsymbol{\delta},\mathbf{z}}}{\partial x_1}\right\rangle v_1(\boldsymbol{\delta},t)+\left\langle f(\mathbf{x},t),\frac{\partial\psi_{\boldsymbol{\delta},\mathbf{z}}}{\partial x_2}\right\rangle v_2(\boldsymbol{\delta},t)= \frac{d}{dt}\left\langle f(\mathbf{x},t),\psi_{\boldsymbol{\delta},\mathbf{z}}\right\rangle+\varepsilon_z(\boldsymbol{\delta},t)\,. \quad (6)$$

where $\langle\ \rangle$ is a inner product sign, and $\varepsilon_z(\boldsymbol{\delta},t)$ is an approximation error inside the scale of $\psi_{\boldsymbol{\delta},\mathbf{z}}$.

The video stream is a sequence of the frames with some inter-frame gap. In up-conversion mode the frame rate is enhanced as much as n_f times, then it is required to find the derivative in the right side of (6) for the moment of time $t=\frac{\Delta i+(i+1)\Delta}{n_f}=(i+\frac{1}{2})\frac{2\Delta}{n_f}$. Let us assume that $n_f=2$. The wavelet optical flow can be found taking into account that the derivative $\frac{d}{dt}\left\langle f(\mathbf{x},t),\psi_{\delta,z}\right\rangle$ in (6) for the instant of time $t=(i+\frac{1}{2})\Delta$ can be determine with the second order error using the finite difference equation

$$\frac{d}{dt}\left\langle f(\mathbf{x},t),\psi_{\boldsymbol{\delta},\mathbf{z}}\right\rangle= \frac{1}{\Delta}\left\langle f(\mathbf{x},(i+1)\Delta)-f_i(\mathbf{x},i\Delta),\psi_{\boldsymbol{\delta},\mathbf{z}}\right\rangle+\varepsilon_\alpha(\boldsymbol{\delta},t)\,. \quad (7)$$

Therefore, (6) with (7) can be rewritten as

$$\left\langle\frac{f(\mathbf{x},i\Delta)+f(\mathbf{x},(i+1)\Delta)}{2},\frac{\partial\psi_{\delta,z}}{\partial x_1}\right\rangle v_1(\boldsymbol{\delta},t)+ \left\langle\frac{f(\mathbf{x},i\Delta)+f(\mathbf{x},(i+1)\Delta)}{2},\frac{\partial\psi_{\delta,z}}{\partial x_2}\right\rangle v_2(\boldsymbol{\delta},t)= \quad (8)$$

$$\frac{1}{\Delta}\left\langle f(\mathbf{x},(i+1)\Delta)-f(\mathbf{x},i\Delta),\psi_{\delta,z}\right\rangle+\varepsilon_z(\boldsymbol{\delta},t)+\varepsilon_\alpha(\boldsymbol{\delta},t)\,,$$

where $\varepsilon_\alpha(\delta,t)$ is the second order error.

Two last terms, i.e., $\varepsilon_z(\delta,t)+\varepsilon_\alpha(\delta,t)$, can be neglected when either the velocity is almost unchangeable inside the appropriate wavelet scale, or, at least, the velocity is small enough. Hence, as (8) shows, finding of the projection of the optical flow to the multi-resolution wavelet allows us instead of the velocity vector estimation inside the large scales, to estimate it precisely inside the sequence of the gently diminished scales as Fig. 1 shows. As a consequence, the matching problem can be solved with less error [12].

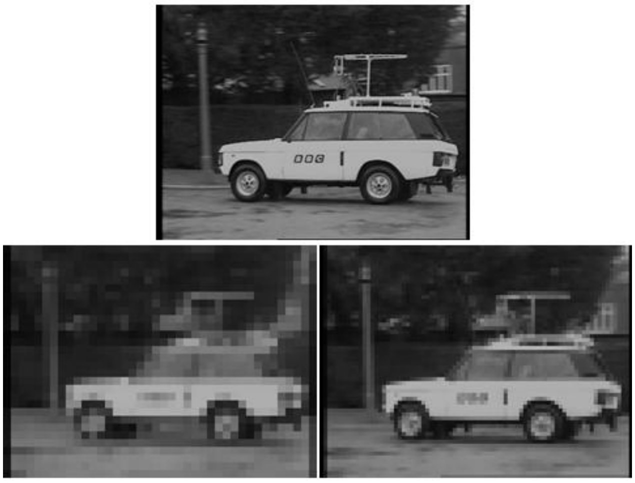

Fig. 1. The original image and two its different scale multi-resolution approximation.

IV. Testing of the Algorithm

The proposed algorithm was tested for the video stream with the various sort of motion. To avoid the object aliasing the non-homogeneous double cross search of the spatial-temporal frame information in the high frequency domain is used. Two reference frames, *i*-th and (*i*+1)-th are used, and between them the predicted frame is inserted, i.e. n_f = 2. In the case when the velocity vectors of two reference frames do not coincide, the (*i*-1)-th frame is used as an auxiliary reference frame to obtain some additional information. For the reference frames two steps of discrete wavelet transform is processed. Firstly, the size of the matching blocks is defined in the scale [-2; 2]. The closed object velocities in the low frequency domain of the second level, i.e. LL_2, are searched (Fig. 2). Then, the appropriate blocks in the searching area are matched. For the matching, only those blocks are chosen, whose velocity vectors between $(i-1)$-th and $(i+1)$-th frames is larger as much as twice than the velocity vector between *i*-th and $(i+1)$-th frames.

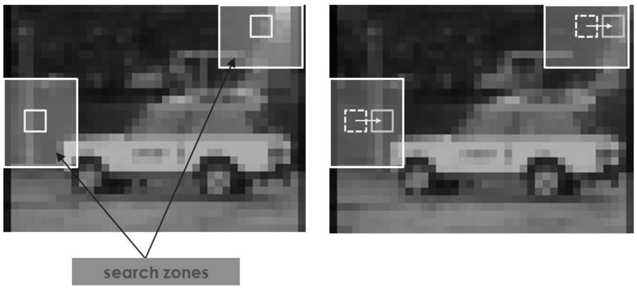

Fig. 2. Search of motion vectors in the search zone.

The choosing blocks are flagged as a significant. Afterward, the velocity vectors are searched in the area [-8; +8] of the significant blocks for the law frequency domain, i.e. LL_1, and then defined more precisely in the domains HL_1, LH_1, HH_1 and LL_0. After wavelet decomposition the next steps are implemented.

A. Determination of the search zone

1. Finding approximations in the low frequency domain.
2. Delete the incorrect motion vectors.
3. Build a search zone as Fig. 3 shows.

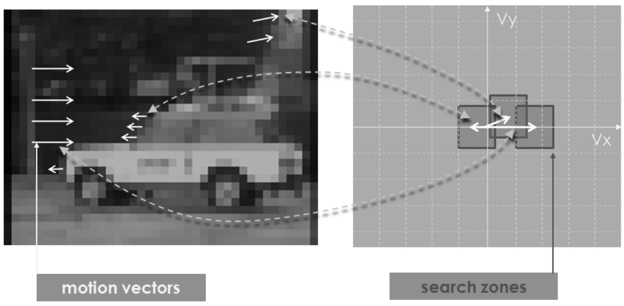

Fig. 3. Search zone is the area 8x8 pixels around ends of vectors

B. Motion detection

4. Find the approximations in the low frequency domain using determined search zone.
5. Accurate motion vectors based on low frequency domain and original image.

C. Determination of the visible overlap areas

6. Determine the visible areas.
7. Determine the overlap and error areas.
8. Set motion vectors for overlapping areas.

D. Checking of the motion vectors

9. Detect the incorrect regions based on incorrect motion vectors (Fig. 4).

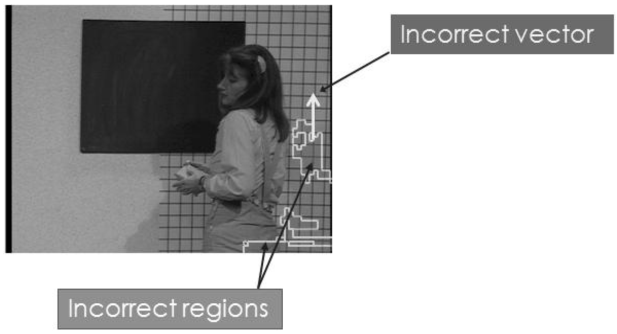

Fig. 4. Detection of the incorrect regions

10. Build a regions map.
11. Delete the incorrect motion vectors.

V. Conclusion

The algorithm will be use in mobile video within spatial frame decimation and interpolation. This technology allows increase a transfer rate at the least eight times due to the fact that information content was reduced.

Acknowledgment

This work was supported by a grant from ISCT.

References

[1] J.B. Lee, and H. Kalva, *The VC-1 and H.264 Video Compression Standards for Broadband Video Services,* Springer, 2008.

[2] A. G. Bors¸ and I. Pitas, "Prediction and Tracking of Moving Objects in Image Sequences," *IEEE Trans. Image Proc.*, vol. 9, no. 8, Aug. 2000.

[3] J. Weber and J. Malik, "Robust computation of optical flow in a multi-scale differential framework," *Int. Jour. of Computer Vision,* vol. 14, pp. 5-19, 1995.

[4] Ch. P. Bernard, "Discrete Wavelet Analysis for Fast Optic Flow Computation," *Applied and Computational Harmonic Analysis,* vol. 11, no. 1, pp. 32-63, July 2001.

[5] J.L. Barron, D.J. Fleet, and S.S. Beauchemin, "Performance of optical flow techniques," *Int. Jour. on Computer Vision*, vol. 12, no. 1, pp. 43-77, 1994.

[6] R. Zahiri-Azar, and S. E. Salcudean, "Motion estimation in ultrasound images using time domain cross correlation with prior estimates," *IEEE Transactions on Biomedical Engineering,* vol. 53, no. 10, Oct. 2006.

[7] V. Patil, and R. Kumar,"A Fast Inverse Motion Compensation Algorithm for DCT-Domain Video Transcoder," *IEEE Trans. on Circuits And Systems For Video Technology*, vol. 18, no. 3, March 2008

[8] Chun-Ho Cheung, and Lai-Man Po. "A Novel Cross-Diamond Search Algorithm for Fast Block Motion Estimation", *IEEE Trans. Circuits And Systems For Video Technology*, vol. 12, no. 12, pp. 1168-1177, Dec. 2002.

[9] M. Cagnazzo, F. Castaldo, T. André, M. Antonini, and M. Barlaud, "Optimal motion estimation for wavelet motion compensated video coding," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 17, no. 7, July 2007.

[10] H.-W. Park, and H.-S. Kim. "Motion estimation using low-band-shift method for wavelet based moving-picture coding," *IEEE Trans. Image Process*., vol.9, no.4, pp. 577–587, April 2000.

[11] Yu Liu, and King Ngi Ngan. "Fast multiresolution motion estimation algorithms for wavelet-based scalable video coding", *Signal Processing: Image Communication,* vol. 22, no. 5, pp. 448-46, June 2007.

[12] V. Moroz, and V. Zaharov. "Temporal Video Sequence Interpolation Based on Motion Compensation in Wavelet Domain", *The 2009 World Congress In Computer Science, Computer Engineering and Applied Computing*, ISBN: 1-60132-092-2/ CSREA Press – Las Vegas, Nevada, USA, July 13-16, 2009.