A STATISTICAL APPROACH TO MOTION VECTOR FIELD SMOOTHING FOR BLOCK-BASED MOTION-COMPENSATED FRAME INTERPOLATION

Dooseop Choi[⋆], *Wonseok Song*[⋆], *Jongsoon Park*[⋆], *Hyuk Choi*[†], *and Taejeong Kim*[⋆]

[⋆] School of Electrical Engineering and Computer Sciences and
Institute of New Media and Communications,
Seoul National University, South Korea
[†] School of Computer Science, University of Seoul, South Korea
Email: dschoi@infolab.snu.ac.kr

ABSTRACT

This paper proposes a new approach to motion vector field smoothing for block-based motion-compensated frame interpolation (MCFI). Based on the assumption that an observed motion vector field, which is the result of a block-based motion estimation (BME), is a degraded version of the true motion vector field, we calculate the *maximum a posteriori* (MAP) estimate of the true motion vector field from the observed. The degradation and the true motion vector field are modeled as additive Gaussian noise and a Markov random field, respectively. Iterative conditional modes (ICM) method is used for calculating the MAP estimate. The experimental results show that the proposed algorithm not only smoothes MVFs but also preserves motion boundaries better than the existing methods.

Index Terms— Motion vector field smoothing, MCFI, MAP-MRF

1. INTRODUCTION

Motion-compensated frame interpolation (MCFI) has been used to enhance temporal resolution of a video by increasing the frame rate. It usually consists of two major steps: motion estimation (ME) and frame interpolation (FI). In the ME step, the motion vectors of non-overlapping blocks or pixels of the current frame are estimated based on the surrounding frames. Block-matching approaches have widely been used for ME because of its simplicity. In the FI step, one or more frames are interpolated using the surrounding frames and the estimated motion vectors.

MCFI that uses block-based motion estimation (BME) suffers from annoying blocky artifacts and ghost effects. This is because BME usually estimates motion vectors by only minimizing the block-matching errors between adjacent frames. One of the solutions to this problem is motion vector field smoothing, which many researches have adopted and improved.

One way to produce a smooth motion vector field (MVF) is to adjust each motion vector in relation to its neighboring vectors. A widely used method is vector median filtering (VMF) [1]. Alparone et al. proposed an adaptively weighted vector median filter where the matching errors are used to calculate the weights [2]. Huang et al. smoothed the MVF according to a measure of reliability that is computed for each vector from both the matching error and the correlations between neighboring motion vectors [4][5]. Sohn et al. proposed a regularization algorithm for motion vector field smoothing [3]. The method iteratively smoothes the MVF based on the matching error and the variance of the MVF. In this paper, we propose a novel statistical approach to motion vector field smoothing for block-based motion-compensated frame interpolation.

2. PROBLEM FORMULATION

Given two adjacent frames $\mathbf{f}_{n-1}$ and $\mathbf{f}_n$ of a video, let $\mathbf{d}_n = [\mathbf{d}_n(1), ..., \mathbf{d}_n(N)]$ denotes the 2-D motion vector field (MVF) obtained by a block-based motion estimation (BME), where $\mathbf{d}_n(i)$ and N respectively denote the 2-D motion vector of the i-th block of the frame $\mathbf{f}_n$ pointing to a matching block in $\mathbf{f}_{n-1}$ and the number of non-overlapping blocks in a frame. Let's assume that $\mathbf{d}_n$ which we call the *observed motion vector field*, or simply *observation* is a degraded version of the true motion with an additive noise such that

$$\mathbf{d}_n(i) = \mathbf{u}_n(i) + \mathbf{e}_n(i), \;\; i = 1, 2, ..., N, \tag{1}$$

where $\mathbf{u}_n(i)$ is the true 2-D motion vector of the i-th block of $\mathbf{f}_n$ and $\mathbf{e}_n(i)$ denotes a random noise. Under this assumption, we want to estimate the true MVF $\mathbf{u}_n = [\mathbf{u}_n(1), ..., \mathbf{u}_n(N)]$ from the *observation* $\mathbf{d}_n$ by maximizing the *a posteriori probability*

$$P(\mathbf{u}_n \mid \mathbf{d}_n, \mathbf{f}_{n-1}, \mathbf{f}_n) = \frac{P(\mathbf{d}_n \mid \mathbf{u}_n, \mathbf{f}_{n-1}, \mathbf{f}_n) P(\mathbf{u}_n \mid \mathbf{f}_{n-1}, \mathbf{f}_n) P(\mathbf{f}_{n-1}, \mathbf{f}_n)}{P(\mathbf{d}_n, \mathbf{f}_{n-1}, \mathbf{f}_n)}. \tag{2}$$

Therefore, the maximum *a posteriori* (MAP) estimate $\hat{\mathbf{u}}_n$ of $\mathbf{u}_n$ can be obtained by

$$\hat{\mathbf{u}}_n = \underset{\mathbf{u}_n}{\operatorname{argmax}} [P(\mathbf{d}_n \mid \mathbf{u}_n, \mathbf{f}_{n-1}, \mathbf{f}_n) \; P(\mathbf{u}_n \mid \mathbf{f}_{n-1}, \mathbf{f}_n)], \tag{3}$$

where $P(\mathbf{d}_n, \mathbf{f}_{n-1}, \mathbf{f}_n)$ and $P(\mathbf{f}_{n-1}, \mathbf{f}_n)$ are ignored because they are constants with respect to the unknown.

2.1. Observation Likelihood

In the literature, the difference between the true motion vector and the motion vector estimated by BME is usually modeled as having generalized Gaussian distribution [6]. Therefore, we assume that the 2-D random noise vectors $\mathbf{e}_n(i)$ in (1) are independent Gaussian with $\mathcal{N}(\mathbf{0}, \sigma_i^2 \mathbf{I})$, where $\mathbf{0}$ and $\mathbf{I}$ are the 2×1 zero vector and the 2×2 identity matrix, respectively, and σ_i^2 is the noise variance. With the assumption that $p(\mathbf{d}_n(i) \mid \mathbf{u}_n, \mathbf{f}_{n-1}, \mathbf{f}_n) = p(\mathbf{d}_n(i) \mid \mathbf{u}_n(i))$ and our observation model (1), we can express the likelihood $p(\mathbf{d}_n(i) \mid \mathbf{u}_n, \mathbf{f}_{n-1}, \mathbf{f}_n)$ as

$$\begin{aligned} &p(\mathbf{d}_n(i) \mid \mathbf{u}_n, \mathbf{f}_{n-1}, \mathbf{f}_n) \\ &= \frac{1}{2\pi\sigma_i^2} \exp\Big(- \frac{\|\mathbf{d}_n(i) - \mathbf{u}_n(i)\|^2}{2\sigma_i^2} \Big). \end{aligned} \quad (4)$$

Finally, conditional independence assumption leads us to the observation likelihood $P(\mathbf{d}_n \mid \mathbf{u}_n, \mathbf{f}_{n-1}, \mathbf{f}_n)$ in the following form:

$$\begin{aligned} &P(\mathbf{d}_n \mid \mathbf{u}_n, \mathbf{f}_{n-1}, \mathbf{f}_n) = \Pi_{i=1}^N p(\mathbf{d}_n(i) \mid \mathbf{u}_n(i)) \\ &= \exp\Big(- N \ln 2\pi - \sum_{i=1}^N \ln \sigma_i^2 - \sum_{i=1}^N \frac{\|\mathbf{d}_n(i) - \mathbf{u}_n(i)\|^2}{2\sigma_i^2} \Big). \end{aligned} \quad (5)$$

2.2. A priori distribution of the true motion vector field

Let us assume that $\mathbf{u}_n$ is a realization of a 2-D Markov random vector field $\mathbf{U}_n$ so that $P(\mathbf{u}_n \mid \mathbf{f}_{n-1}, \mathbf{f}_n)$ follows the Gibbs distribution

$$P(\mathbf{u}_n \mid \mathbf{f}_{n-1}, \mathbf{f}_n) = \frac{1}{Z} \exp\Big(- \gamma U(\mathbf{u}_n) \Big), \quad (6)$$

where $Z = \sum_{\mathbf{u}_n \in \mathbb{D}} \exp\Big(- \gamma U(\mathbf{u}_n) \Big)$, $\mathbb{U}$ is a set of all possible configurations of $\mathbf{U}_n$, $U(\cdot)$ is the energy function for the priori distribution, and γ is a constant. The new energy function $U(\mathbf{u}_n)$ is defined as

$$U(\mathbf{u}_n) = \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} \mu_{ij} \|\mathbf{u}_n(i) - \mathbf{u}_n(j)\|^2, \quad (7)$$

where $\mathcal{N}_i$ denotes the index set of the blocks neighboring on the i-th block and Figure 1-(a) shows the neighborhood system. The factor μ_{ij} is the weight for the neighboring motion vector $\mathbf{u}_n(j)$ of the i-th block. The proposed energy function is designed to reflect the fact that the MVFs of a common video are smooth except at the motion boundaries, where the weight μ_{ij} plays a crucial role.

This is how we compute μ_{ij}. Let $\varepsilon_i(\mathbf{v})$ be the sum of absolute differences (SAD), as the block-matching error corresponding to the motion vector $\mathbf{v}$, such that, with $\mathbf{f}_n(\mathbf{m})$ denoting the pixel value at the position $\mathbf{m} = [m_x \ m_y]$ in the frame $\mathbf{f}_n$,

$$\varepsilon_i(\mathbf{v}) = \sum_{\mathbf{m} \in \mathcal{B}_i} |\mathbf{f}_n(\mathbf{m}) - \mathbf{f}_{n-1}(\mathbf{m} + \mathbf{v})|, \quad (8)$$

where $\mathcal{B}_i$ denotes the set of all the pixel indices in the i-th block. Also, let us define the set $\mathcal{N}_i^x = \{j | j \in \mathcal{N}_i, x_i(j) = 1\}$ of block indices, which we call the *suitable set*, where $x_i(j)$ is given by

$$x_i(j) = \begin{cases} 1, & \varepsilon_i(\mathbf{u}_n(j)) \le \lambda \\ 0, & \varepsilon_i(\mathbf{u}_n(j)) > \lambda \end{cases} \quad (9)$$

with a threshold λ. We choose $\lambda = \min\{\lambda_1, \lambda_2\}$ where $\lambda_1 = \frac{1}{2}(\max(\mathcal{S}) + \min(\mathcal{S}))$, $\lambda_2 = \mathrm{median}(\mathcal{S})$, and $\mathcal{S} = \{\varepsilon_i(\mathbf{u}_n(k)) \mid k \in \mathcal{N}_i\}$. Then the proposed weight factor μ_{ij} is defined as $\mu_{ij} = (1 - \zeta)\alpha_{ij} + \zeta\beta_{ij}$ where

$$\alpha_{ij} = \frac{\varepsilon_i(\mathbf{u}_n(j))^{-1} \cdot x_i(j)}{\sum_{k \in \mathcal{N}_i^x} \varepsilon_i(\mathbf{u}_n(k))^{-1}},$$

$$\beta_{ij} = \frac{[\sum_{l \in \mathcal{N}_i^x, l \neq j} \|\mathbf{u}_n(j) - \mathbf{u}_n(l)\|^2]^{-1} \cdot x_i(j)}{\sum_{j \in \mathcal{N}_i^x} [\sum_{l \in \mathcal{N}_i^x, l \neq j} \|\mathbf{u}_n(j) - \mathbf{u}_n(l)\|^2]^{-1}},$$

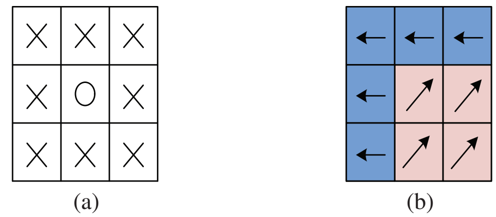

Fig. 1. (a) Second-order neighborhood system. $\circ$ and $\times$ indicate a center block and neighboring blocks, respectively. (b) An example of a motion boundary. The blocks of the same color correspond to the same motion.

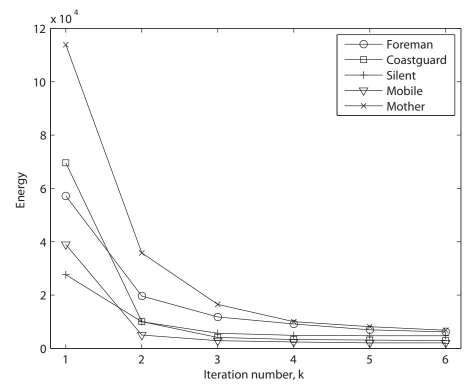

Fig. 2. The total energy $E(\hat{\mathbf{u}}_n^k)$ versus iteration number k.

and ζ is a constant with $0 \le \zeta \le 1$. Note that $0 \le \mu_{ij} \le 1$ and $\sum_{j \in \mathcal{N}_i} \mu_{ij} = 1$ because $\sum_{j \in \mathcal{N}_i} \alpha_{ij} = \sum_{j \in \mathcal{N}_i} \beta_{ij} = 1$.

The proposed weight factor is devised to show 'how *suitable* and *reliable* a neighboring motion vector is for the current block'. We can use Figure 1-(b) to explain how it works. In the figure, the blocks with the same color correspond to the same motion. The three lower right neighboring vectors are in this case considered suitable candidates for the true motion of the center block, and these neighbors should be given higher weights in calculating $U(\mathbf{u}_n)$. Since the true motion vector must have the smallest possible SAD, we use SAD to define the *suitability* measure α_{ij}. On the other hand, SAD might be misleading when a very similar pattern occurs in a wrong place. This is counteracted by the *reliability* measure β_{ij}, which gives a higher weight to the most representative vector in the *suitable set* by incorporating the distances between motion vectors in the set.

3. ENERGY MINIMIZATION

The maximization of the *a posteriori probability* is equivalent to the minimization of the total energy

$$\begin{aligned} E(\mathbf{u}_n) = \sum_{i=1}^N V(\mathbf{u}_n(i)) = \sum_{i=1}^N \Big[&\frac{\|\mathbf{d}_n(i) - \mathbf{u}_n(i)\|^2}{2\sigma_i^2} + \ln \sigma_i^2 \\ &+ \gamma \sum_{j \in \mathcal{N}_i} \mu_{ij} \|\mathbf{u}_n(i) - \mathbf{u}_n(j)\|^2 \Big] \end{aligned} \quad (10)$$

Fig. 3. Refined motion vector fields. (a) The observed motion vector field (b) VMF [1] (c) adaptive VMF [2] (d) vector regularization with 6 iterations [3] (e) Proposed with 3 iterations (f) Proposed with 6 iterations.

Fig. 4. Interpolated frames using refined motion vector fields. (a) The observed motion vector field (b) VMF [1] (PSNR=29.27dB) (c) adaptive VMF [2] (PSNR=29.35dB) (d) vector regularization with 6 iterations [3] (PSNR=28.81dB) (e) Proposed with 3 iterations (PSNR=29.37dB) (f) Proposed with 6 iterations (PSNR=29.62dB).

because the a posteriori probability can be rewritten as

$$P(\mathbf{u}_n \mid \mathbf{d}_n, \mathbf{f}_{n-1}, \mathbf{f}_n) \propto \exp\Big(-\sum_{i=1}^{N} V(\mathbf{u}_n(i))\Big). \tag{11}$$

Because $V(\mathbf{u}_n(i))$ is a convex function of $\mathbf{u}_n(i)$, we can minimize $E(\mathbf{u}_n)$ by minimizing each of $V(\mathbf{u}_n(i))$ for $i = 1, ..., N$. Let $\hat{\mathbf{u}}_n(i)$ denote the vector that minimizes $V(\mathbf{u}_n(i))$. Then $\hat{\mathbf{u}}_n(i)$ can be found by solving $\frac{\partial V(\mathbf{u}_n(i))}{\partial \mathbf{u}_n(i)} = 0$, which gives rise to

$$\hat{\mathbf{u}}_n(i) = \frac{\mathbf{d}_n(i) + 2\gamma\sigma_i^2 \sum_{j\in\mathcal{N}_i} \mu_{ij}\mathbf{u}_n(j)}{1 + 2\gamma\sigma_i^2}. \tag{12}$$

Here, we propose adapting (12) into the following iterative procedure because, $\{\mathbf{u}_n(j) \mid j \in \mathcal{N}_i\}$ and σ_i^2 being unknown, it does not produce a direct solution. Finally, we note that μ_{ij} is regarded as a constant while solving $\frac{\partial V(\mathbf{u}_n(i))}{\partial \mathbf{u}_n(i)} = 0$ since it is not a function of $\mathbf{u}_n(i)$.

Step 1) Calculate an initial estimate $\hat{\mathbf{u}}_n^0$ of $\mathbf{u}_n$ from

$$\hat{\mathbf{u}}_n^0(i) = \frac{\mathbf{d}_n(i) + 2\gamma\sigma_{i(0)}^2 \sum_{j\in\mathcal{N}_i} \mu_{ij}\mathbf{d}_n(j)}{1 + 2\gamma\sigma_{i(0)}^2}, \tag{13}$$

where $\sigma_{i(0)}^2$ is an approximate initial estimate of the noise variance derived from the set $\{\mathbf{d}_n(l) \mid l \in \{i \cup \mathcal{N}_i^x\}\}$.

Step 2) Calculate the k-th estimate $\hat{\mathbf{u}}_n^k$ from $\hat{\mathbf{u}}_n^{k-1}$ by

$$\hat{\mathbf{u}}_n^k(i) = \frac{\mathbf{d}_n(i) + 2\gamma\sigma_{i(k-1)}^2 \sum_{j\in\mathcal{N}_i} \mu_{ij}\hat{\mathbf{u}}_n^{k-1}(j)}{1 + 2\gamma\sigma_{i(k-1)}^2}, \tag{14}$$

where $\sigma_{i(k-1)} = \|\mathbf{d}_n(i) - \hat{\mathbf{u}}_n^{k-1}(i)\|$.

Step 3) Iterate Step 2 while increasing k until $|E(\hat{\mathbf{u}}_n^k) - E(\hat{\mathbf{u}}_n^{k-1})| \leq \epsilon$ is satisfied for a pre-defined threshold ϵ.

We note that the proposed iterative algorithm falls in the category of iterative conditional modes (ICM) method [8]. Therefore, $E(\mathbf{u}_n)$ being a convex function of $\mathbf{u}_n$, the algorithm should certainly converge to a global optimum if σ_i, $i = 1, ..., N$, are known.

4. EXPERIMENTAL RESULTS

In this section, we describe our experiments and results intended to demonstrate the performance of the proposed algorithm. Our

Table 1. Averaged PSNR(dB) Performance

Sequences	VMF [1]	WVMF [2]	VR (iter=6) [3]	Proposed (iter=3)	Proposed (iter=6)
Foreman	27.56	27.60	27.18	27.78	$\mathbf{27.81}$
Coastguard	28.99	29.57	28.71	29.86	$\mathbf{29.90}$
Silent	28.43	28.40	28.20	28.59	$\mathbf{28.66}$
Mobile	20.97	20.81	20.09	20.97	$\mathbf{21.04}$
Mother	28.30	28.24	25.73	28.36	$\mathbf{28.36}$

method is compared with the vector median filter (VMF) [1], the adaptive vector median filter (AVMF) [2], and the vector regularization method (VR) [3]. For the experiments, five commonly used test videos, *Foreman*, *Coastguard*, *Silent*, *Mobile*, and *Mother* are used. Each video is composed of 300 frames of size 352×288 pixels at 24 fps. A full-search block matching algorithm is performed to obtain the observed MVFs between two successive odd frames with the block size 8×8 pixels over the search range +/-24 pixels both in horizontal and vertical directions. The sum of absolute difference (SAD) is used for the matching criterion. We set $\gamma = 5$ and $\zeta = 0.5$ for the experiments.

We first tested the convergence of the proposed algorithm. The backward MVF from 19-th frame to 17-th frame of the five test videos are used. Figure 2 shows the energy $E(\hat{\mathbf{u}}_n^k)$ versus the iteration number k. We can see that the energy $E(\hat{\mathbf{u}}_n^k)$ quickly converges to a minimum value in only 3∼4 iterations.

The subjective quality of the interpolated frames constructed by different methods are presented in Figure 3. Here we show, overlaid, the backward MVFs pointing from 121-th frame to 119-th frame of the video *Silent*. A woman in the video is moving only her right arm so that large motion vectors are observed on her right arm and its the shadow on her shoulder. The figure shows that the proposed method not only well smoothes the MVF but also preserves the motion boundaries (see the motion vectors of the finger, the arm, and the shadow) through iterative smoothing of the field. This is because the proposed algorithm exploits only the neighborhood motion vectors that are “suitable” for the current block to refine the motion vector. In contrast, the methods in [1] and [3] spoil the motion boundaries. Note also that the method in [2], too, well preserves the motion boundaries, which is because Alparone et al. use as well the concept of suitability in assigning the weights to the vector median filter.

Figure 4 shows the 120-th frame interpolated using the refined MVFs of each algorithm. In generating the interpolated frame, we adopted the method of [7], which uses both forward and backward MVFs. We can see in the figure that the proposed algorithm shows the best performance. Especially, clearer edges can be seen around the arm and its shadow as compared to the results of the other algorithms.

Finally, table 1 shows the averaged PSNR for each of the five test videos. The PSNR is calculated from the error between the interpolated and original even numbered frame and then averaged over the whole video. We find that the proposed algorithm clearly shows the over-all best performance. The proposed method particulary shows good performance on the sequences with many motion boundaries such as *Foreman* and *Coastguard*. On the sequences with only few motion boundaries such as *Mother* and *Silent*, however, the improvement is not as great.

5. CONCLUSION

A new algorithm for motion vector field smoothing is proposed in this paper. Under the MAP-MRF framework, the true motion vector field (MVF) is estimated based on the assumption that the observed MVF, as a result of a block-based ME, is a degraded version of the true MVF. The experimental results show that the proposed algorithm not only smoothes MVFs but also preserves motion boundaries better than the existing methods.

6. ACKNOWLEDGMENT

This work is supported by Brain Korea 21 project 2011. This work (Grants No.00045359-2) also was supported by Business for Cooperative R&D between Industry, Academy, and Research Institute funded Korea Small and Medium Business Administration in 2011.

7. REFERENCES

[1] J. Astola, P. Haavisto, and Y. Neuvo, “Vector median filters,” *Proceedings of the IEEE*, Vol. 78, No. 4, pp. 678-689, April 1990.

[2] L. Alparone, M. Barni, F. Bartolini, and V. Cappellini “Adaptively weighted Vector-median filters for motion fields smoothing,” *Proceedings of ICASSP*, Vol. 4, pp. 678-689, April 1990.

[3] Y. W. Sohn, and M. G. Kang “Block-based motion vector smoothing for periodic pattern region,” *ICIAR 2007*, LNCS 4633, pp. 491 500, 2007.

[4] A.-M. Huang, and T. Q. Nguyen “A multistage motion vector processing method for motion-compensated frame interpolation,” *IEEE Transactions on Image Processing*, Vol. 17, No. 5, pp. 694-708, May 2008.

[5] A.-M. Huang, and T. Q. Nguyen “Correlation-based motion vector processing with adaptive interpolation scheme for motion-compensated frame interpolation,” *IEEE Transactions on Image Processing*, Vol. 18, No. 4, pp. 740-752, April 2009.

[6] G. Dane, and T. Q. Nguyen “The efficienty of motion-compensating prediction for hybrid coding of video sequences,” *IEEE Journal on Selected Areas in Communications*, Vol. 5, Issue. 7, pp. 1140-1154, August 1987.

[7] D. Wang, A. Vincent, P. Blanchfield, and R. Klepko “Motion-compensated frame rate up conversion - Part 2: New algorithms for frame interpolation,” *IEEE Transactions on Broadcasting*, Vol. 56, No. 2, pp. 142-149, June 2010.

[8] J. Besag “On the statistical analysis of dirty pictures,” *Journal of the Royal Statistical Society*, Series B 48, pp. 259-302, 1986.