Hindawi Publishing Corporation
Advances in Fuzzy Systems
Volume 2013, Article ID 652429, 9 pages
http://dx.doi.org/10.1155/2013/652429

Research Article

Coding B-Frames of Color Videos with Fuzzy Transforms

Ferdinando Di Martino and Salvatore Sessa

Università degli Studi di Napoli Federico II, Dipartimento di Architettura, Via Toledo 402, 80134 Napoli, Italy

Correspondence should be addressed to Salvatore Sessa; sessa@unina.it

Received 24 May 2013; Accepted 26 June 2013

Academic Editor: Irina Perfilieva

We use a new method based on discrete fuzzy transforms for coding/decoding frames of color videos in which we determine dynamically the GOP sequences. Frames can be differentiated into intraframes, predictive frames, and bidirectional frames, and we consider particular frames, called Δ-frames (resp., R-frames), for coding P-frames (resp., B-frames) by using two similarity measures based on Lukasiewicz t-norm; moreover, a preprocessing phase is proposed to determine similarity thresholds for classifying the above types of frame. The proposed method provides acceptable results in terms of quality of the reconstructed videos to a certain extent if compared with classical-based F-transforms method and the standard MPEG-4.

1. Introduction

A video can be considered as a sequence of frames of sizes $N \times M$; a frame is an image that can be compressed by using a lossy compression method. We can classify each frame as intraframe (for short, I-frame), predictive frame (for short, P-frame), and bidirectional frame (for short, B-frame) which is more compressible than I-frame. A B-frame can be predicted or interpolated from an earlier and/or later frame. In order to avoid a growing propagation error, a B-frame is not used as a reference to make further predictions in most encoding standards except in AVC [1]. A frame can be considered as a P-frame if it is "similar" to the previous I-frame in the frame sequence; otherwise, it must be considered as a new I-frame. This similarity relation between a P-frame and the previous I-frame is fundamental in video-compression processes because a P-frame has values in its pixels very close to the pixels of the previous I-frame. This suggests to define a frame containing differences between a P-frame and the previous I-frame, called Δ-frame which has a low quantity of information and hence it can be coded with a low compression rate. A P-frame is decoded via the previous I-frame and the Δ-frame. In the MPEG-4 method [2, 3], that adopts the JPEG technique [4] for coding/decoding frames, the I-frames, P-frames, and B-frames are arranged in a Group of Picture (for short, GOP) sequence. A B-frame is reconstructed by using either the previous or successive I-frame. Here the results of [5] are improved by using a technique based on F-transforms for coding B-frames. For convenience, we assume that the first frame of a video is an I-frame. We assign an ID number to each frame of the video. Then we can say that the kth frame is a B-frame or a P-frame if it is "very similar" to the previous ith I-frame in the sense that its similarity $\mathrm{Sim}(i, k)$ a parameter defined on the Lukasiewicz t-norm (see formula (12)) is greater than a threshold value $\mathrm{Sim}P$ [5]; otherwise the kth frame is assumed to be a new I-frame as the first frame of the successive GOP sequence.

The first algorithm is used for determining the GOP sequences; the second algorithm is used for determining the type of P-frame or B-frame. The first frame of the GOP sequence is always an I-frame and the last frame is a P-frame. The function "analyze GOP sequence (ID1, ID2)" reported in Algorithm 1 describes this process, where ID1 is the ID of the first I-frame and ID2 is the ID of the last P-frame in the GOP sequence. This function is used for determining if the kth frame in the GOP sequence, where $\mathrm{ID1} < k < \mathrm{ID2}$, is a B-frame or a P-frame. We define a threshold similarity $\mathrm{Sim}B$, and we compare it with the frame whose ID is formed from the integer $[Ms]$ contained in the mean Ms of the previous I-frame or P-frame and the kth frame by obtaining a similarity value $\mathrm{Sim}(k, [Ms])$. In the array element $\mathrm{NP}[k]$ we insert the ID number of the last frame after the kth frame for which

Sim(k, [Ms]) < Sim$B$ holds. The variable i contains the ID number of the previous I-frame or P-frame; it is initially called ID1; the variable w points to the last frame in the GOP sequence; it is called ID2.

Algorithm 1 (analyze GOP sequence (ID1, ID2)). Pseudo-code for determining a GOP sequence

(1) i = ID of the first I-frame //i is the ID of first frame of the video

(2) w = number of frames //w is the ID of the last P-frame of the video

(3) $k = i + 1$

(4) IF $k < w$

(5) Calculate the similarity Sim(i, k) between the kth frame and the ith frame

(6) If Sim$(i, k) <$ Sim$P$,

- (a) the kth frame is a B-frame or a P-frame and is inserted in the GOP sequence
- (b) $k = k + 1$

(7) Else

- (a) analyse GOP sequence $(i, k - 1)$
- (b) $i = k$
- (c) go to (3)

(8) End.

Algorithm 2. Pseudo-code for determining type of frames

(1) i = ID1//i is the ID of the first frame of the GOP sequence

(2) w = ID2//w is the ID of the last P-frame of the GOP sequence

(3) For each k in $[i + 1, w - 1]$
NP[k] = k//Initialize NP[k]

(4) $s = k + 1$

(5) Create the [Ms]th frame as a new frame whose normalized pixels are obtained as the mean between the normalized pixels of the ith and sth frames

(6) Calculate the similarity Sim(k, [Ms]) between the kth and [Ms]th frames. If Sim(k, [Ms]) < Sim$B$,

- (a) NP[k] = $s - 1$
- (b) Else $s = s + 1$
- (c) go to step (6)

(7) next for

(8) NPMin = min(NP[k])

(9) The frames between the ith and NPMin-th frames are labelled as B-frames

(10) The NPMin-th frame is labelled as a P-frame

(11) If NPmin < w then

- (a) i = NPMin,
- (b) go to step (2)

(12) End.

In our approach we determine a GOP sequence at each step. The frame after the last P-frame is the I-frame of the new GOP sequence. After determining the GOP sequences of the color video, we use the F-transforms [5, 7–10] for compressing the frames. The F-transform method has been developed in [5]. In this paper each frame is converted in the YUV space. Indeed, since the human eye perceives an image mostly in the Y band (brightness) with respect to the U and V bands (chrominance), we can use a stronger compression rate for coding the image in U and V bands with respect to that one used for coding the image in the Y band, without loss of information in the reconstructed image. In [5] the authors show that the quality of the reconstructed images is better than the one obtained using the F-transform method directly in the RGB space (see also [11, 12]). The proposed method is widely discussed in Section 4. In Sections 2 and 3 the theory of F-transforms and its application are recalled for image compression, respectively. In Section 5 the results are deduced on a large color videos dataset.

2. Fuzzy Transforms

We recall from [9] some essential definitions. Let $n \geq 3$ and $x_1, x_2, \ldots, x_n$ be points (nodes) of $[a, b]$ such that $x_1 = a < x_2 < \cdots < x_n = b$. The fuzzy sets $A_1, \ldots, A_n : [a, b] \to [0, 1]$ form a fuzzy partition of $[a, b]$ if

(1) $A_i(x_i) = 1$ for any $i = 1, 2, \ldots, n$;

(2) $A_i(x) = 0$ if $x \notin (x_{i-1}, x_{i+1})$, where $i = 1, 2, \ldots, n$ and $x_0 = x_1 = a$, $x_{n+1} = x_n = b$;

(3) $A_i(x)$ is a continuous function on $[a, b]$;

(4) $A_i(x)$ is strictly increasing on the interval $[x_{i-1}, x_i]$ for $i = 2, \ldots, n$ and is strictly decreasing on the interval $[x_i, x_{i+1}]$ for $i = 1, \ldots, n - 1$;

(5) for any $x \in [a, b]$, $\sum_{i=1}^{n} A_i(x) = 1$.

We say that $\{A_1, A_2, \ldots, A_n\}$ constitute a symmetric fuzzy partition if the following hold:

(6) equidistance of the nodes, that is, $x_i = a + h \cdot (i - 1)$ for $i = 1, 2, \ldots, n$, where $h = (b - a)/(n - 1)$;

(7) $A_i(x_i - x) = A_i(x_i + x)$ for any $x \in [0, h]$ and $i = 2, \ldots, n - 1$;

(8) $A_{i+1}(x) = A_i\,(x - h)$ for any $x \in [x_i, x_{i+1}]$ and $i = 1, 2, \ldots, n - 1$.

Considering functions f taking values on a finite set $P = \{p_1, \ldots, p_m\} \subseteq [a, b]$, $f : P \to [0, 1]$, we suppose that P is sufficiently dense with respect to a fuzzy partition $\{A_1, A_2, \ldots, A_n\}$ of $[a, b]$, that is, if $m > n$ and for each $i = 1, \ldots, n$ there exists an index $j \in \{1, \ldots, m\}$ such that $A_i(p_j) > 0$. Now let $n, m \geq 3$, $y_1, y_2, \ldots, y_m \in [c, d]$ be other m assigned nodes such that $y_1 = c < \cdots < y_m = d$. Let $C_1, \ldots, C_m : [c, d] \to [0, 1]$ be another fuzzy partitions of $[c, d]$. Let $f : P \times Q \to [0, 1]$ be a function defined on the finite set $P \times Q = \{p_1, \ldots, p_N\} \times \{q_1, \ldots, q_M\} \subseteq [a, b] \times [c, d]$, with $N > n$ and $M > m$, where P (resp., Q) is sufficiently dense with respect to some fuzzy partition

$\{A_1, A_2, \ldots, A_n\}$ of $[a, b]$ (resp., $\{C_1, \ldots, C_m\}$ of $[c, d]$). Then $[F_{kl}]$, $F_{kl} \in [0,1]$, $k = 1, \ldots, n$ and $l = 1, \ldots, m$, is the fuzzy matrix which is defined as discrete F-transform of f with respect to $\{A_1, A_2, \ldots, A_n\}$ and $\{C_1, \ldots, C_m\}$ if the following holds:

$$F_{kl} = \frac{\sum_{j=1}^{M} \sum_{i=1}^{N} f\left(p_i, q_j\right) A_k\left(p_i\right) C_l\left(q_j\right)}{\sum_{j=1}^{M} \sum_{i=1}^{N} A_k\left(p_i\right) C_l\left(q_j\right)}. \quad (1)$$

Afterwards we define $f_{nm}^F : P \times Q \to [0,1]$ to be the inverse F-transform of f with respect to $\{A_1, A_2, \ldots, A_n\}$ and $\{C_1, \ldots, C_m\}$ as

$$f_{nm}^F\left(p_i, q_j\right) = \sum_{k=1}^{n}\sum_{l=1}^{m} F_{kl} A_k\left(p_i\right) C_l\left(q_j\right). \quad (2)$$

The following theorem holds.

Theorem 3. *Let $f : P \times Q \to [0,1]$ be a function assigned on $P \times Q = \{p_1, \ldots, p_N\} \times \{q_1, \ldots, q_M\} \subseteq [a, b] \times [c, d]$. Then for every $\varepsilon > 0$, there exist two integers $n(\varepsilon)$, $m(\varepsilon)$ with $n(\varepsilon) < N$, $m(\varepsilon) < M$ and some fuzzy partitions $\{A_1, A_2, \ldots, A_{n(\varepsilon)}\}$ of $[a, b]$ and $\{C_1, C_2, \ldots, C_{m(\varepsilon)}\}$ of $[c, d]$ for which P and Q are sufficiently dense with respect to these partitions, respectively, and such that the following inequality holds for every $i = 1, \ldots, N$, $j = 1, \ldots, M$:*

$$\left| f\left(p_i, q_j\right) - f_{n(\varepsilon)m(\varepsilon)}^F\left(p_i, q_j\right) \right| < \varepsilon. \quad (3)$$

3. The Coding/Decoding Process

Let R be an image of sizes $N \times M$, considered as a fuzzy relation $R : (i, j) \in \{1, \ldots, N\} \times \{1, \ldots, M\} \to [0,1]$; that is, $R(i, j) = P(i, j)/L_t$, with $P(i, j)$ being the normalized value of the pixel with respect to the length Lt of the scale used. For simplicity, let $p_i = i$, $q_j = j$, $a = c = 1$, $b = N$, and $d = M$. Let the fuzzy sets $A_1, \ldots, A_n : [1, N] \to [0,1]$ and $C_1, \ldots, C_m : [1, M] \to [0,1]$, with $n < N$ and $m < M$, form a fuzzy partition of $[1, N]$ and $[1, M]$, respectively. Following [8], R is subdivided in submatrices R_B of sizes $N(R_B) \times M(R_B)$, $R_B : (i, j) \in \{1, \ldots, N(R_B)\} \times \{1, \ldots, M(R_B)\} \to [0,1]$, called blocks, coded to matrices of sizes $n(R_B) \times m(R_B)$, $(n(R_B) < N(R_B), m(R_B) < M(R_B))$ via the following discrete F-transforms $[F_{kl}^B]$ for every $(k, l) \in \{1, \ldots, n(R_B)\} \times \{1, \ldots, m(R_B)\}$ as

$$F_{kl}^B = \frac{\sum_{j=1}^{M(R_B)} \sum_{i=1}^{N(R_B)} R_B(i, j) A_k(i) C_l(j)}{\sum_{j=1}^{M(R_B)} \sum_{i=1}^{N(R_B)} A_k(i) C_l(j)}, \quad (4)$$

and decode $[F_{kl}^B]$ via $R_{n(R_B)m(R_B)}^F : (i, j) \in \{1, \ldots, N(R_B)\} \times \{1, \ldots, M(R_B)\} \to [0,1]$ defined as

$$R_{n(R_B)m(R_B)}^F = \sum_{j=1}^{M(R_B)} \sum_{i=1}^{N(R_B)} F_{kl}^B A_k(i) C_l(j) \quad (5)$$

which approximates R_B in the sense of Theorem 3; that is, there exist, for every $\varepsilon > 0$, two integers $n(R_B, \varepsilon)$, $m(R_B, \varepsilon)$ such that the following holds for every $(i, j) \in \{1, \ldots, N(R_B)\} \times \{1, \ldots, M(R_B)\}$:

$$\left| R_B(i, j) - R_{n(R_B,\varepsilon)m(R_B,\varepsilon)}^F(i, j) \right| < \varepsilon. \quad (6)$$

Unfortunately the previous theorem does not suggest a method for finding such integers, and then we try to assign values to $n(R_B) = n(R_B, \varepsilon)$ and $m(R_B) = m(R_B, \varepsilon)$ for getting compression rates given by

$$\rho(R_B) = \frac{n(R_B) \cdot m(R_B)}{N(R_B) \cdot M(R_B)} \quad (7)$$

which are useful to code any original block R_B. The recomposition of the blocks $R_{n(R_B)m(R_B)}^F$ gives the image R^F whose PSNR with respect to the original image R is calculated via the following well-known formula:

$$\operatorname{PSNR}\left(R, R^F\right) = 20\log_{10} \frac{Lt}{\sqrt{\sum_{i=1}^{N} \sum_{j=1}^{M} \left(R(i, j) - R^F(i, j)\right)^2 / N \times M}}. \quad (8)$$

In accordance with [8], in the proposed experiments the best results are deduced with the symmetric fuzzy partitions $A_1, \ldots, A_{n(R_B)} : [1, N(R_B)] \to [0,1]$ and $C_1, \ldots, C_{m(R_B)} : [1, M(R_B)] \to [0,1]$ defined as

$$A_1(i) = \begin{cases} 0.5\left(\cos\frac{\pi}{h}(i-1)+1\right) & \text{if } 1 \le i \le x_2, \\ 0 & \text{else}, \end{cases}$$

$$A_k(i) = \begin{cases} 0.5\left(\cos\frac{\pi}{h}(i-x_k)+1\right) & \text{if } x_k \le i \le x_{k+1}, \\ 0 & \text{else}, \end{cases}$$

$$A_{n(R_B)}(i) = \begin{cases} 0.5\left(\cos\frac{\pi}{h}\left(i-x_{n(R_B-1)}\right)+1\right) & \text{if } x_{n(B)-1} \le i \le N(R_B), \\ 0 & \text{else}, \end{cases} \quad (9)$$

where $k = 2, \dots, n(R_B) - 1$, $h = (N(R_B) - 1)/(n(R_B) - 1)$, $x_k = 1 + h \cdot (k - 1)$, and

$$C_1(j) = \begin{cases} 0.5\left(\cos\frac{\pi}{s}(j-1)+1\right) & \text{if } 1 \le j \le y_2, \\ 0 & \text{else,} \end{cases}$$
$$C_t(j) = \begin{cases} 0.5\left(\cos\frac{\pi}{s}(j-y_t)+1\right) & \text{if } y_{t-1} \le j \le x_{t+1}, \\ 0 & \text{else,} \end{cases}$$
$$C_{m(R_B)}(j) = \begin{cases} 0.5\left(\cos\frac{\pi}{s}\left(j-y_{m(R_B)-1}\right)+1\right) & \text{if } y_{m(R_B)-1} \le j \le M(R_B), \\ 0 & \text{else,} \end{cases} \quad (10)$$

where $t = 2, \dots, m(R_B) - 1$, $s = (M(R_B) - 1)/(m(R_B) - 1)$, and $y_t = 1 + s \cdot (t - 1)$.

4. Our Proposal

The proposed process includes the following steps:

(1) each color frame, seen as a fuzzy relation, is converted from the space *RGB* to the space *YUV*;

(2) a classification of the frames is made via the previous algorithms;

(3) the compression rate $\rho_I = \rho_I(R_B)$ of the I-frames is the mean of three (possibly different) compression rates used in the three bands, that is, if any block R_B of an I-frame has sizes (say) $N_{IY}(R_B) \times M_{IY}(R_B)$ in the band Y and is coded to a block of sizes (say) $n_{IY}(R_B) \times m_{IY}(R_B)$ for which the related compression rate is given by $\rho_{IY} = \rho_{IY}(R_B) = (n_{IY}(R_B) \cdot m_{IY}(R_B)) \cdot (N_{IY}(R_B) \cdot M_{IY}(R_B))^{-}$ and the analogous meaning has the symbols ρ_{IU}, ρ_{IV}. Of course we have $\rho_I = (\rho_{IY} + \rho_{IU} + \rho_{IV})/3$. A similar meaning can be given to $\rho_\Delta = \rho_\Delta(R_B)$ (resp., $\rho_R = \rho_R(R_B)$) for Δ-frames (resp., R-frames).

A color image in the *RGB* space with pixels normalized in $[0, 1]$ is converted to *YUV* space via the formula [5]

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.332 & 0.500 \\ 0.500 & -0.419 & -0.0813 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 0.5 \\ 0.5 \end{bmatrix}. \quad (11)$$

Since no misunderstanding can arise, a frame is denoted by a capital letter instead of its ID number in a sequence of a video. In step (2), the similarity measure adopted in [5] is used for classifying the type of frame. It is based on the Lukasiewicz t-norm between two frames F and G, with $F, G : (i, j) \in \{1, 2, \dots, N\} \times \{1, 2, \dots, M\} \to [0, 1]$, defined as

$$\operatorname{Sim}(F,G) = \left(\sum_{i=1}^{N}\sum_{j=1}^{M}\{1 - \max\{F(i,j), G(i,j)\} + \min\{F(i,j), G(i,j)\}\}\right) \times (N \times M)^{-1}. \quad (12)$$

In the μth band ($\mu \in \{Y, U, V\}$) we will use the symbol $\operatorname{Sim}_\mu(F, G)$. The authors [5] have shown that Lukasiewicz t-norm provides the best results with respect to other t-norms as the classical Min and the arithmetical product. For convenience, we assume that the first frame of a video is an I-frame. For determining a GOP sequence in a single band, it can be verified if the successive frame G is a B-frame or a P-frame, that is, if it is "very similar" to the preceding I-frame F in the sense that $\operatorname{Sim}(F, G) < \operatorname{Sim}B$, with $\operatorname{Sim}B \in [0, 1]$ being a prefixed threshold value; otherwise G is assumed to be a new I-frame. We determine a GOP sequence in an assigned band using (12) with the following process:

(1) we consider the first frame F as an I-frame;

(2) we compare F with the successive frame G;

(3) if $\operatorname{Sim}(F, G) < \operatorname{Sim}P$, the frame G is a B-frame or a P-frame and is enclosed in the GOP sequence. Then we consider the successive frame G and go to step (2); otherwise G is a new I-frame. The previous frame is a P-frame and represents the last frame of the GOP sequence.

After determining the GOP sequence, we check if each frame of the sequence is a B-frame or a P-frame by using the previous algorithms. In step (3) we finally compress the frames. In order to reduce the mean compression rate for a P-frame, in [5] and references therein, the authors introduce a "difference" frame D, called Δ-frame, between a P-Frame G and I-frame F by defining $D : (i, j) \in \{1, 2, \dots, N\} \times \{1, 2, \dots, M\} \to [0, 1]$ as

$$D(i,j) = \frac{[F(i,j) - G(i,j) + 1]}{2}. \quad (13)$$

The usage of the Δ-frame has the advantage of using a stronger compression rate for the P-frames with respect to the I-frames; indeed a P-frame G has values in its pixels very close to the pixels of the previous I-frame. Hence the Δ-frame D in (13) has a low quantity of information and it can be coded with a low compression rate. Then, if D' and F' are the frames obtained after coding/decoding F and D, the frame G' (reconstruction of the frame G), with $D', F', G' : (i, j) \in \{1, 2, \dots, N\} \times \{1, 2, \dots, M\} \to [0, 1]$, is deduced from the membership values of F' and D' via the following formula:

$$G'(i,j) = \frac{\max\{0, F'(i,j) - 2D'(i,j) + 1\}}{\max\{1, F'(i,j) - 2D'(i,j) + 1\}}. \quad (14)$$

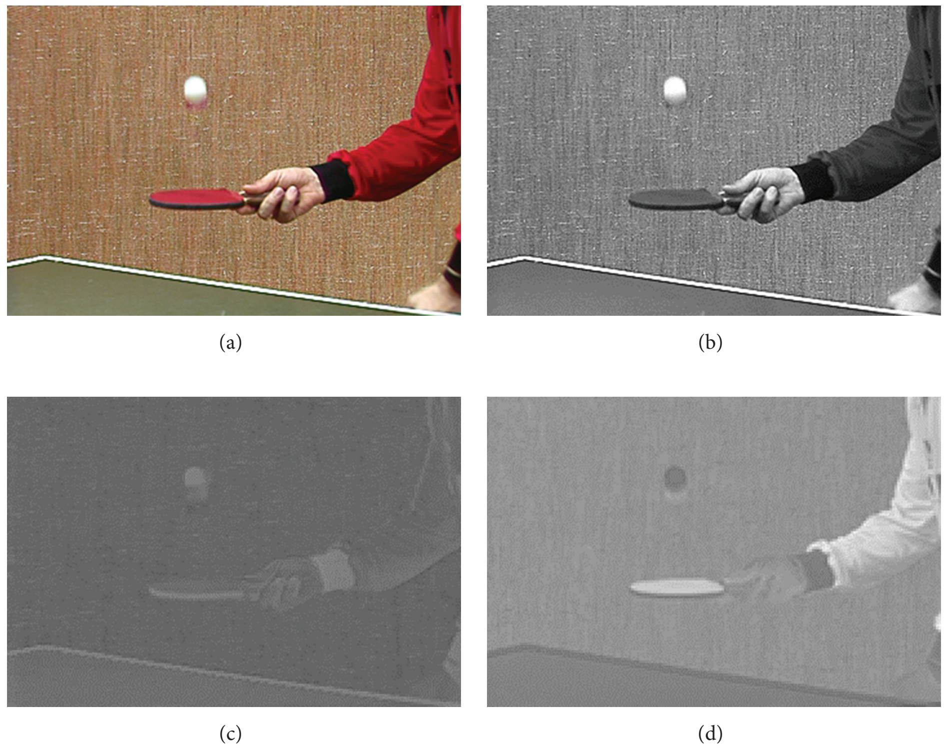


Figure 1: (a) Frame 1 of "tennis2" [6], (b) Frame 1 in Y band, (c) Frame 1 in U band, and (d) Frame 1 in V band.

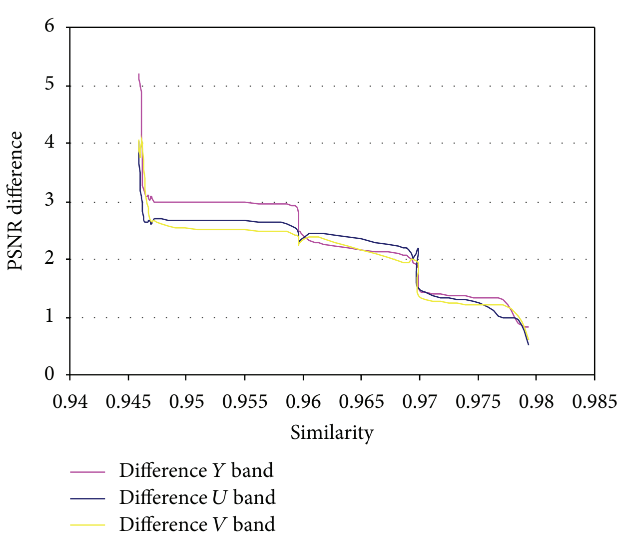


Figure 2: Diff(PSNR) with the similarity in Y, U, and V bands.

Now we present a new schema for coding/decoding a B-frame which is inserted in a GOP between an I-frame F and a P-frame G. Then we consider a frame R given by

$$R(i,j) = \frac{[(F(i,j) + G(i,j))/2 - B(i,j) + 1]}{2} \tag{15}$$

and we code it. Let R' be the frame obtained after decoding R, with $R' : (i,j) \in \{1,2,\ldots,N\} \times \{1,2,\ldots,M\} \rightarrow [0,1]$. All the coding/decoding processes are realized via the F-transforms with the symmetric fuzzy partition given in Section 3. We reconstruct the B-frame, say B', by combining the membership values of F', G', and R' via the following formula:

$$B'(i,j) = \frac{\max\left\{0, \left[F'(i,j) + G'(i,j)\right]/2 - 2R'(i,j) + 1\right\}}{\max\left\{1, \left[F'(i,j) + G'(i,j)\right]/2 - 2R'(i,j) + 1\right\}}. \tag{16}$$

We use the formulas (14) and (16) for reconstructing the P-frames and the B-frames in the videos, respectively. In accordance with [5], we convert each image in the RGB space by using the formula

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.4075 \\ 1 & -0.3455 & -0.7169 \\ 1 & 1.7790 & 0 \end{bmatrix} \begin{bmatrix} Y \\ U \\ V \end{bmatrix} + \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \end{bmatrix}. \tag{17}$$

For simplicity of presentation, in our tests here we adopt $M(R_B) = N(R_B)$, $m(R_B) = n(R_B)$. In [5] a preprocessing phase is adopted for determining the threshold Sim$P$ calculated with the following steps:

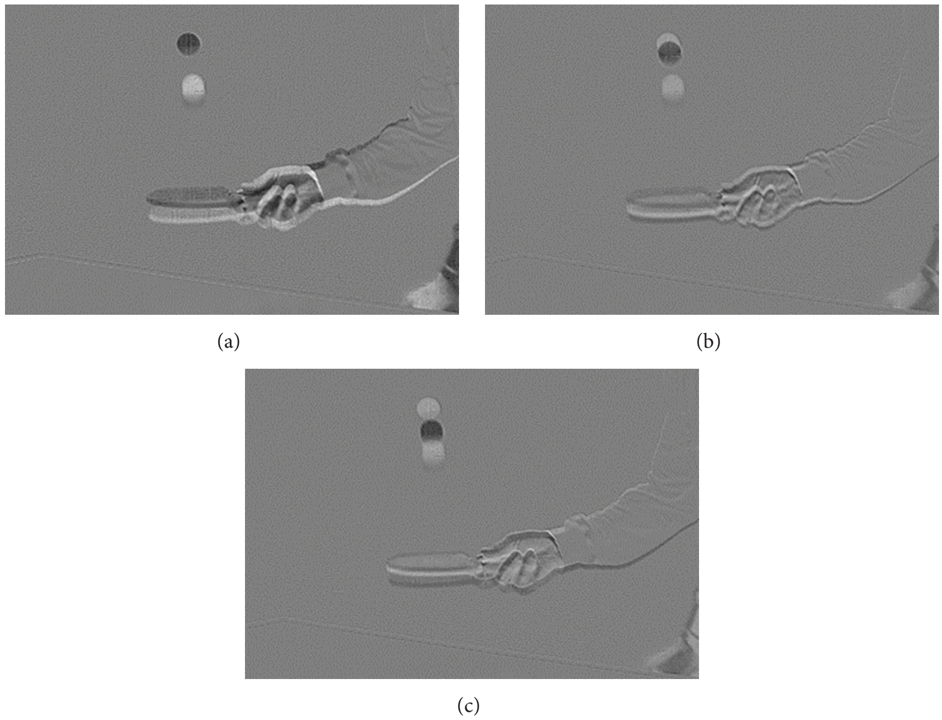


FIGURE 3: (a) Δ-frame from Frame 4 in Y band, (b) R-frame from Frame 2 in Y band, and (c) R-frame from Frame 3 in Y band.

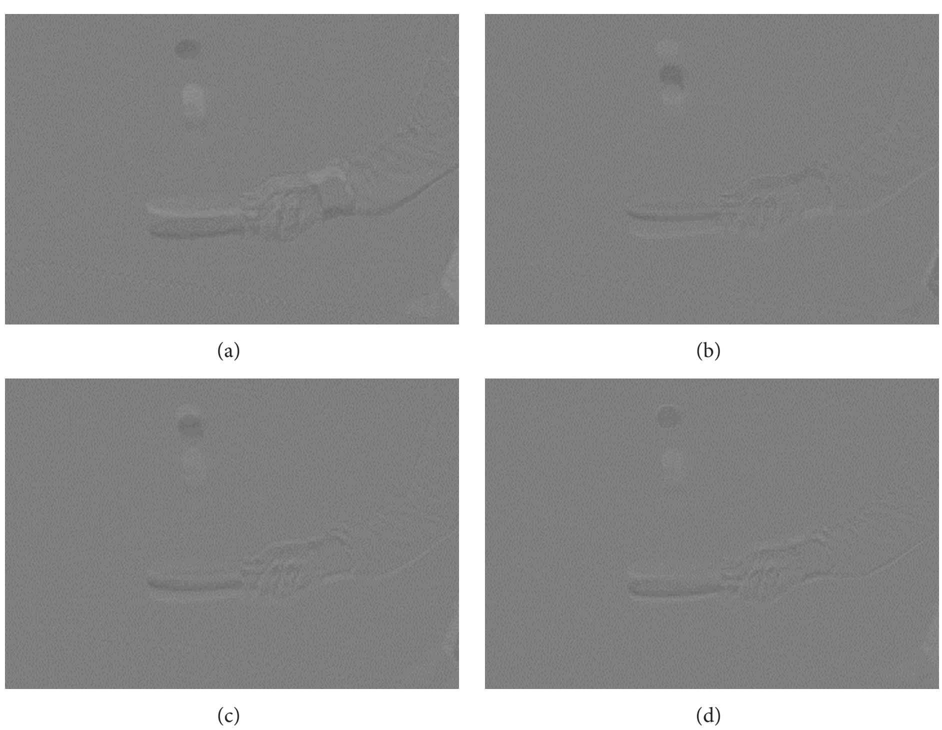


FIGURE 4: (a) Δ-frame from Frame 6 in U band, (b) R-frame from Frame 2 in U band, (c) R-frame from Frame 3 in U band, and (d) R-frame from Frame 4 in U band.

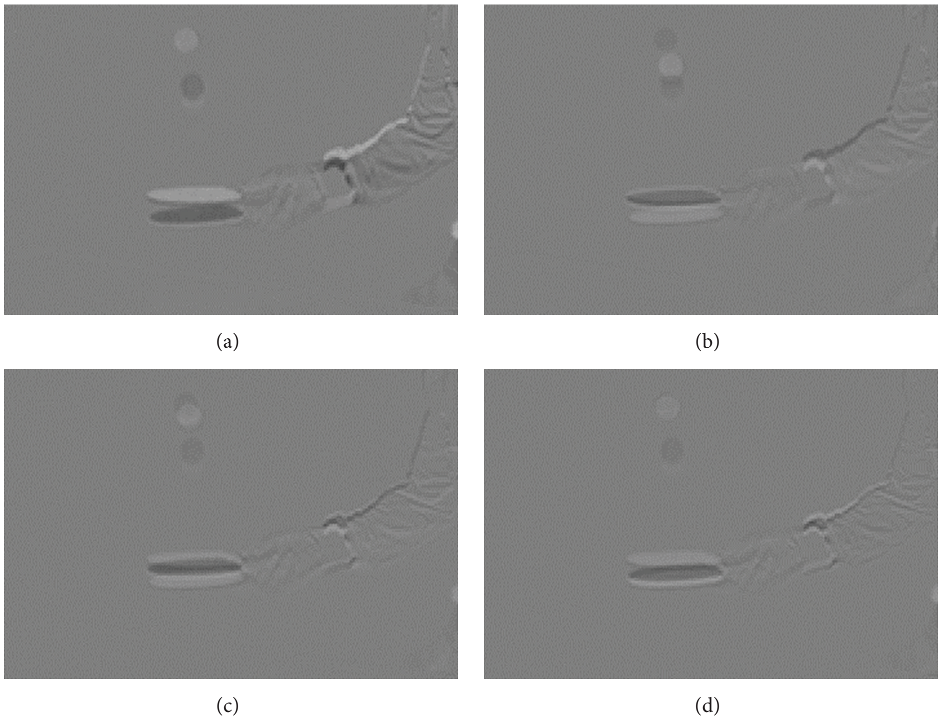


Figure 5: (a) Δ-frame from Frame 5 in V band, (b) R-frame from Frame 2 in V band, (c) R-frame from Frame 3 in V band, and (d) R-frame from Frame 4 in V band.

(1) if the initial frame F is considered as an I-frame, we compress F in the μth band ($\mu \in \{Y, U, V\}$) with compression rate $\rho_{I\mu}$; each successive frame is a P-frame G and we archive the similarity value $\mathrm{Sim}_\mu(F, G)$ calculated with formula (12); we compress the Δ-frame D in the μth band with compression rate equal to $\rho_{P\mu}$ (less than of $\rho_{I\mu}$) and if D' is the related decompressed frame, we derive the P-frame G' via (14);

(2) each P-frame G is also coded in the μth band with compression rate $\rho_{P\mu}$ and let G'' be the decoded P-frame by using directly the F-transforms; then we determine the difference diff(PSNR) $= |\mathrm{PSNR}(G'', G) - \mathrm{PSNR}(G', G)|$;

(3) the trend of diff(PSNR) is plotted with respect to the similarity $\mathrm{Sim}_\mu(F, G)$ in each band of the image. As similarity threshold, we assume that value of $\mathrm{Sim}_\mu(F, G)$ such that diff(PSNR) does not exceed a prefixed limit is equal to 3 (cf. [5] for details);

(4) then the threshold Sim$P$ is given by

$$\mathrm{Sim}P = \max_{G \in \mathrm{GOP}} \left\{ \max \left\{ \mathrm{Sim}_\mu (F, G) : \mu \in \{Y, U, V\} \right\} \right\} \quad (18)$$

with F being the first I-frame of the GOP sequence. In our tests, in addition we put Sim$B$ = Sim$P$ in the preprocessing phase.

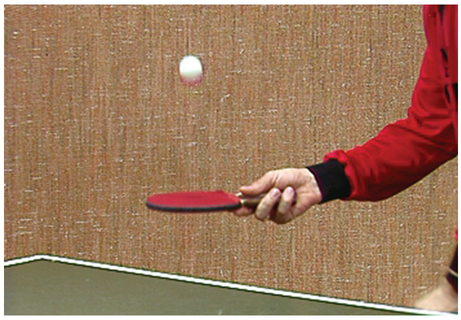
5. The Results

For brevity of discussion, we show the results obtained for the color video "tennis2" [6]. We present all the results by assuming $\rho_I \approx 0.262$ for the I-frames, $\rho_\Delta \approx 0.027$ for the Δ-frames, and $\rho_R \approx 0.020$ for the R-frames. Figures 1(a)–1(d) show the first frame of the video and the corresponding single-band images in the YUV space, respectively. As example of Diff(PSNR), Figure 2 contains the plots of Diff(PSNR) ≤ 3 for the similarity values obtained in Y, U, and V bands for which we choose $\mathrm{Sim}_Y(F, G) > 0.948 = \mathrm{Sim}P$ (as average). As examples we show some Δ-frames and R-frames in each band.

(i) *Y Band.* The first P-frame is given by the fourth frame. Figure 3(a) contains the Δ-frame obtained by using (13) from the fourth frame and the first frame (an I-frame). The second and the third frames are B-frames. Figure 3(b) (resp., Figure 3(c)) shows the R-frame obtained by using (15) from the second (resp., third) frame, the first frame (an I-frame), and the fourth frame (a P-frame).

(ii) *U Band.* The first P-frame is given by the sixth frame. Figure 4(a) contains the Δ-frame obtained by using (13) from the sixth frame and the first frame (an I-frame). The frames 2, 3, and 4 are B-frames. Figures 4(b)–4(d) show the R-frames obtained by using (15) from the first frame (an I-frame), the B-frames 2, 3, and 4, and the sixth frame (a P-frame), respectively.

(iii) *V Band.* The first P-frame is given by the fifth frame. Figure 5(a) contains the Δ-frame obtained by using (13) from



(a)

(b)

(c)

Figure 6: (a) Frame 2 in the proposed method, (b) Frame 2 in F-transforms, and (c) Frame 2 in MPEG-4.

Table 1: Results for "tennis2" [6] in the proposed method.

Parameters	Y band	U band	V band
Number of I-frames	15	7	8
Number of P-frames	31	23	23
Number of B-frames	54	70	69
Mean compression rate $\rho(B)$	0.1128	0.0236	0.0245
Mean PSNR for I-Frames	27.011	25.545	25.812
Mean PSNR for P-Frames	24.816	23.710	23.815
Mean PSNR for B-Frames	24.734	22.819	23.026

the sixth frame and the first frame (an I-frame). The frames 2, 3, and 4 are B-frames. Figures 5(b)–5(d) show the R-frames obtained by using (15) from the first frame (an I-frame), the B-frames 2, 3, and 4, and the fifth frame (a P-frame), respectively.

All the results obtained for the video "tennis2" are synthetized in Table 1.

Figures 6(a)–6(c) contain Frame 2 decoded with the proposed method, classical F-transforms, and MPEG-4, respectively.

In Table 2 we report the final PSNR index in the three methods.

6. Conclusions

We present a new method for coding/decoding color videos, in which we classify a frame in I-frame, P-frame, and B-frame using similarity measures for determining the GOP sequences and the type of frames. Our method seems to be fully comparable with classical F-transforms and MPEG-4 for similar mean compression rates to a certain extent.

Table 2: Comparison with other methods for "tennis2" [6].

Parameters	Proposed method	F-transforms	MPEG-4
Mean compression rate	0.053	0.058	0.055
Mean PSNR	23.915	22.801	23.431

Acknowledgments

The authors thank the referees and the editor whose suggestions have greatly improved the contents of this paper.

References

[1] I. E. G. Richardson, *The H. 264 Advanced Videos Compression Standard*, John Wiley & Sons, Hoboken, NJ, USA, 2010.

[2] F. C. N. Pereira and T. Ebrahimi, *MPEG-4 Book*, Prentice Hall Professional, Upper Saddle River, NJ, USA, 2002.

[3] T. Sikora, *MPEG: Digital Video Coding Standards, Digital Electronics Handbook*, McGraw- Hill, New York, NY, USA, 1995.

[4] W. B. Pennebaker and J. L. Mitchell, *JPEG: Still Im Age Data Compression Standard*, Van Nostrand Reinhold, New York, NY, USA, 1993.

[5] F. Di Martino, V. Loia, and S. Sessa, "Fuzzy transforms for compression and decompression of color videos," *Information Sciences*, vol. 180, no. 20, pp. 3914–3931, 2010.

[6] http://sampl.eng.ohio-state.edu/~sampl/database.htm.

[7] F. Di Martino, V. Loia, I. Perfilieva, and S. Sessa, "An image coding/decoding method based on direct and inverse fuzzy transforms," *International Journal of Approximate Reasoning*, vol. 48, no. 1, pp. 110–131, 2008.

[8] F. Di Martino and S. Sessa, "Compression and decompression of images with discrete fuzzy transforms," *Information Sciences*, vol. 177, no. 11, pp. 2349–2362, 2007.

[9] I. Perfilieva, "Fuzzy transforms: theory and applications," *Fuzzy Sets and Systems*, vol. 157, no. 8, pp. 993–1023, 2006.

[10] I. Perfilieva and B. De Baets, "Fuzzy transforms of monotone functions with application to image compression," *Information Sciences*, vol. 180, no. 17, pp. 3304–3315, 2010.

[11] H. Nobuhara, W. Pedrycz, and K. Hirota, "Relational image compression: optimizations through the design of fuzzy coders and YUV color space," *Soft Computing*, vol. 9, no. 6, pp. 471–479, 2005.

[12] H. Nobuhara, W. Pedrycz, S. Sessa, and K. Hirota, "A motion compression/reconstruction method based on max t-norm composite fuzzy relational equations," *Information Sciences*, vol. 176, no. 17, pp. 2526–2552, 2006.