Research Article

Frame Interpolation Based on Visual Correspondence and Coherency Sensitive Hashing

Lingling Zi,^{1,2} Junping Du,¹ and Qian Wang¹

¹ Beijing Key Laboratory of Intelligent Telecommunication Software and Multimedia, School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China

² School of Electronic and Information Engineering, Liaoning Technical University, Huludao 125105, China

Correspondence should be addressed to Junping Du; junpingdu@126.com

Received 12 March 2013; Accepted 16 June 2013

Academic Editor: Chengjin Zhang

Copyright © 2013 Lingling Zi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The technology of frame interpolation can be applied in intelligent monitoring systems to improve the quality of surveillance video. In this paper, a region-guided frame interpolation algorithm is proposed by introducing two innovative improvements. On the one hand, a detection approach is presented based on visual correspondence for detecting the motion regions that correspond to attracted objects in video sequences, which can narrow the prediction range of interpolated frames. On the other hand, spatial and temporal mapping rules are proposed using coherency sensitive hashing, which can obtain more accurate predicted values of interpolated pixels. Experiments show that the proposed method can achieve encouraging performance in terms of visual quality and quantitative measures.

1. Introduction

Frame interpolation plays a very important role in intelligent monitoring systems. This technology can not only increase the frame rate of surveillance video to meet the requirements of monitoring display devices [1] but also forecast the missing frames to obtain smooth surveillance video. The essence of frame interpolation is to predict intermediate frames by using two given frames in video sequences [2], and it can be roughly divided into two categories: the optical flow method and the block matching method. The former uses time-domain variation and correlation of pixel intensity data to determine the location and value of each pixel [3]. However, in practice, because of the existence of a variety of factors such as multiple light levels, transparency, and noise, assumptions of brightness constancy and spatial smoothness in the optical flow basic equations cannot be always satisfied [4, 5]. The latter finds the best match for each block by minimizing the matching difference [6, 7]. Due to its simplicity, this method is easy to implement, but blocking artifacts will occur in the interpolation frames [8]. In brief, these existing methods still suffer from poor quality of video frames. Therefore, it would

be beneficial to develop an algorithm to obtain high-quality interpolation frames.

In this paper, a novel region-guided frame interpolation algorithm (RGFI) is proposed, the goal of which is to improve the definitions of intermediate frames of the surveillance video. To achieve this, two techniques are introduced for frame interpolation. The first technique is visual correspondence using local descriptor matching [9, 10]. For this purpose, compact real-time descriptors (CARD) [11] are used, an approach which was proposed recently to establish visual correspondence quickly between two images. Moreover, the computation time of each descriptor is approximately 16 times faster than that of the scale-invariant feature transform (SIFT) [12]. The second technique is the latest approximate nearest neighbor (ANN) method [13], called coherency sensitive hashing (CSH) [14], which was proposed to find matching patches quickly. CSH relies on hashing to propagate information through similarity in appearance space and neighborhood in the image plane. Its advantage is the use of special observations to establish candidate block sets, which can avoid many artifacts along edges when reconstructing an original image. Through the advanced techniques described

FIGURE 1: Architecture of the proposed algorithm.

above, the interpolation task for video sequences is achieved satisfactorily from a different perspective.

The main contributions of this paper can be summarized as follows: (1) a novel and comprehensive frame interpolation framework based on spatial and temporal correlations in video sequences and (2) a detection approach based on visual correspondence to capture motion regions so as to narrow the prediction range of interpolation frames. The foundation of the detection approach is the frame difference technique and CARD technique, (3) the spatial and temporal block definition and spatial and temporal mapping relationship definition for better implementation of the RGFI algorithm. On this basis, this approach is combined with CSH to construct spatial and temporal mapping rules so as to predict values of interpolated pixels accurately, (4) application of the proposed algorithm to video resizing. Using accurate motion regions obtained in the proposed algorithm, the important content in the surveillance video can be preserved on the premise of ensuring global visual effect and can be displayed on the monitoring display devices with different resolutions.

The rest of the paper is structured as follows. Section 2 provides an overview of the proposed RGFI algorithm. Section 3 demonstrates the implementation details of RGFI. Section 4 presents experimental work carried out to demonstrate the effectiveness of the algorithm. Section 5 shows another contribution of our algorithm. Section 6 concludes the paper.

2. RGFI Overview

In this paper, a novel RGFI algorithm is proposed according to characteristics of video sequences. The RGFI framework

is shown in Figure 1 and is divided into motion region detection and interpolated pixel computation. First, the detection approach using visual correspondence to capture motion regions is presented. Then the spatial and temporal mapping scheme based on CSH is demonstrated to compute the unknown pixel values in the motion regions obtained. At the same time, for the pixels of other regions of input video frames, only the original values are kept. Through these two steps, high-quality interpolation frames are produced.

3. Implementation of RGFI

3.1. Motion Region Detection Using Visual Correspondence. The motion regions of the surveillance video are very important in the intelligent monitoring system. Therefore, by finding these regions, it is possible to speed up the accomplishment of the interpolation task. Unlike previous methods, the method proposed here takes advantage of visual correspondences based on CARD to obtain motion regions between video frames. The detailed implementation of the detection approach includes three parts: motion region initial estimation, key point correspondence establishment, and motion region determination.

Because the frame difference method [15] can quickly find the outline of the moving target in video sequences, it is first used to estimate rough motion regions, as shown in the following:

$$I_i(x, y) \in M'_n, \quad \text{if } |f_{n+1}(x, y) - f_n(x, y)| \geq D^n \times c, \quad (1)$$

where $f_n(x, y)$ and $f_{n+1}(x, y)$ denote the two consecutive frames of the input video sequences, M'_n is the initial motion region corresponding to $f_n(x, y)$, c is a predefined value, and

D^n is the difference between the maximum value and the minimum value in two consecutive frames as follows:

$$D^n = \max |f_{n+1}(x, y) - f_n(x, y)| - \min |f_{n+1}(x, y) - f_n(x, y)|. \quad (2)$$

Second, inspired by [11], comparatively accurate key point correspondences are established in the initial motion regions; the establishment process includes the following four steps.

Step 1. Construct an image pyramid [16] for the initial regions as follows:

$$L(x_m, y_m, n, \sigma - 1) = L_{2\downarrow}(x_m, y_m, n, \sigma), \quad (3)$$

where $L(x_m, y_m, n, \sigma)$ is a scale function, x_m and y_m are the values of the abscissa and the ordinate, σ is the level of the image pyramid, and $2\downarrow$ is the downsampling factor.

Step 2. A corner detector technique [17] is used to find the key points in the image pyramid, and the corresponding point set is denoted as $K_n = \{k_n(x_{k1}, y_{k1}) \mid (x_{k1}, y_{k1}) \in M'_n\}$. Similarly, the key point set K_{n+1} corresponding to the initial region M'_{n+1} of $f_{n+1}(x, y)$ is obtained.

Step 3. For $\forall k_n(x_i, y_i) \in (K_n \cup K_{n+1})$, the orientation histogram $HG(h'_k)$ is determined; the corresponding magnitudes and orientations are given in (4). At the same time, log-polar binning is used to achieve good discrimination ability [18], as follows

$$\begin{aligned} m_{HG}(x_i, y_i) &= \left((M'_n(x_i + 1, y_i) - M'_n(x_i - 1, y_i))^2 \right. \\ &\quad \left. + (M'_n(x_i, y_i + 1) - M'_n(x_i, y_i - 1))^2 \right)^{1/2}, \quad (4) \\ \omega_{HG}(x_i, y_i) &= \arctg \left(\frac{(M'_n(x_i + 1, y_i) - M'_n(x_i - 1, y_i))}{(M'_n(x_i, y_i + 1) - M'_n(x_i, y_i - 1))} \right), \\ r_{k,n'} &= \begin{cases} 0, & 0 \leq \sqrt{u_i^2 + v_i^2} < \varphi_1 \\ 1 + Q_8 \left(\arctg \left(\frac{u_i}{v_i} \right) - \frac{(2\pi(l_k \bmod L))}{L} \right), & \varphi_1 \leq \sqrt{u_i^2 + v_i^2} < \varphi_2 \\ 9 + Q_8 \left(\arctg \left(\frac{u_i}{v_i} \right) - \frac{(2\pi(l_k \bmod L))}{L} \right), & \varphi_2 \leq \sqrt{u_i^2 + v_i^2} < \varphi_3, \end{cases} \quad (5) \end{aligned}$$

where $r_{k,n'}$ denotes a rotated binning pattern, L is a predefined value, φ_1 , φ_2 , and φ_3 are constants, u_i and v_i denote the coordinate values of $k_n(x_i, y_i)$ in relative coordinates, $Q_F(\theta)$ represents the quantization function, given in (6), and l_k denotes the quantization result of $\omega_{HG}(x_i, y_i)$. On this basis, a spatial binning table [19] is used to extract $HG(h'_k)$ and to

obtain the descriptor d_k^n of $k_n(x_i, y_i)$, setting the descriptor set D_k^n

$$Q_F(\theta) = \left\lfloor \frac{F\theta + \pi}{2\pi} \right\rfloor \bmod F. \quad (6)$$

Step 4. For $\forall d_k^n (d_k^n \in D_k^n)$, compute its corresponding descriptor d_m^{n+1} in the next frame according to the following:

$$\begin{aligned} d_m^{n+1} &= \arg \min_{d_i^{n+1} \in D_k^{n+1}} \text{disc}(d_i^{n+1} - d_k^n) \\ &= \arg \min_{d_i^{n+1} \in D_k^{n+1}} \|d_i^{n+1} - d_k^n\|_1 \\ &= \arg \min_{i \in K_n} \left(\sum_{j=1}^C |\delta_i^{n+1}(j) - \delta_k^n(j)| \right) \quad (7) \\ \delta_k^n &= \frac{(1 + \text{sgn}(W^T d_k^n))}{2}, \end{aligned}$$

where δ_k^n is the changed short binary code of any descriptor d_k^n , C is the length of changed binary code, and W is a weight matrix. In this way, the corresponding set D_m^{n+1} of descriptor set D_k^n and the matching key point K_m of K_n can be obtained.

Finally, more accurate boundary values for the motion regions can be determined according to the locations of key points and their correspondences between consecutive frames. Let $K = K_n \cup K_m$; the boundary values of motion region M^n can then be computed using the following equation:

$$\begin{aligned} M_L^n &= \min_{k_n(x_i^n, y_i^n) \in K} x_i^n - \varepsilon_L, \\ M_R^n &= \max_{k_n(x_i^n, y_i^n) \in K} x_i^n + \varepsilon_R, \\ M_T^n &= \min_{k_n(x_i^n, y_i^n) \in K} y_i^n - \varepsilon_T, \\ M_B^n &= \max_{k_n(x_i^n, y_i^n) \in K} y_i^n + \varepsilon_B, \end{aligned} \quad (8)$$

where M_L^n , M_R^n , M_T^n , and M_B^n are the left border, right border, top border, and bottom border and ε_L , ε_R , ε_T , and ε_B are the predefined minimal deviation values.

Figure 2 shows one example of motion region detection. Figures 2(a) and 2(b) are the two consecutive frames of the walk video, and the key point correspondences between them are presented. From the results for the motion region shown in Figure 2(c), it is evident that the proposed detection approach can perform well in expressing the motion content of video sequences, which lays the foundation for interpolated pixel computation.

3.2. Interpolated Pixel Computation Using Coherency Sensitive Hashing. To facilitate the computation of interpolated pixels, it is necessary to define some concepts in advance.

FIGURE 2: One example of motion region detection. The first two figures from the left show the two consecutive images of the walk video; red lines are used to denote key point correspondences. Marked with yellow lines, the obtained motion region in the last figure contains key points denoted by blue dots.

Definition 1. For obtained motion regions in two consecutive frames, the original motion region M^n and the mapping motion region M^{n+1} are defined. M^n or M^{n+1} is divided into $s \times s$ overlapping image blocks, and each block is defined as a spatial and temporal block, ST for short. Denote the total numbers and sets of ST as T_{M^n} , $T_{M^{n+1}}$, and $B^n = \{b_k^n\}$, $B^{n+1} = \{b_k^{n+1}\}$, respectively.

Definition 2. Given $\forall b_k^n (b_k^n \in B^n)$, assume that $b_l^{n+1} (b_l^{n+1} \in B^{n+1})$ is the corresponding mapping block of b_k^n and that it is computed through spatial and temporal mapping rules. Then $b_k^n \mapsto b_l^{n+1}$ is defined as the spatial and temporal mapping relationship between b_k^n and b_l^{n+1} .

As far as is known, the block matching method generally finds the only match for each original divided block. However, the computation method proposed here can obtain two or more matching blocks for each ST so as to approach more closely the true values of interpolated pixels. This computation method includes four parts: ST projection and conversion, mapping block computation, nearest block determination, and unknown pixel computation.

First, the projection of ST is computed on Walsh-Hadamard kernels [20]. Specifically, for each ST b_j , $b_j \in (B^n \cup B^{n+1})$, gray-code filter kernels [21] are computed as the transform kernel, as shown in (9) and Figure 3. The results are stored in the temporary set $H = \{WHT_j \mid 1 \leq j \leq T^N\}$, in which $T^N = T_{M^n} + T_{M^{n+1}}$:

$$\begin{aligned}
 V_{s1}^{(0)} &= V_{s2}^{(0)} = 1 \\
 V_{s1}^{(k)} &= \left\{ \left[v_{s1}^{(k-1)} \alpha_k v_{s1}^{(k-1)} \right] \right\} \\
 V_{s2}^{(k)} &= \left\{ \left[v_{s2}^{(k-1)} \alpha_k v_{s2}^{(k-1)} \right] \right\} \\
 V_s^{(k)} &= V_{s1}^{(k)} \times V_{s2}^{(k)} \quad \text{s.t. } v_{s1}^{(k-1)} \in V_{s1}^{(k)}, \\
 &\quad v_{s2}^{(k-1)} \in V_{s2}^{(k)}, \quad \alpha_k \in \{+1, -1\}.
 \end{aligned} \tag{9}$$

To accelerate ST mapping, a hash value is assigned to each transform result, and each hash function $h_{a,b}(WH_j) : R^{\dim} \rightarrow N$ maps a $\dim$ -dimensional vector WH_j onto the set of integers [22]. In this way, the corresponding hash table t_m ($1 \leq m \leq T^N$) is constructed, and each ST is saved in the entry $t_m[g_m(b_j)]$, in which $g_m(b_j)$ is given by the following:

$$\begin{aligned}
 g_m(b_j) &= \frac{WHT_1 \cdot b_j + \rho_1}{\gamma} \circ \frac{WHT_2 \cdot b_j + \rho_2}{\gamma} \circ \dots \circ \frac{WHT_j \cdot b_j + \rho_j}{\gamma} \\
 &\quad \circ \dots \circ \frac{WHT_{T^N} \cdot b_j + \rho_{T^N}}{\gamma}.
 \end{aligned} \tag{10}$$

In (10), γ is a fixed integer value which is set in advance and ρ_j is a random number according to a uniform distribution on the interval $[0, \gamma)$. At the same time, for the two vectors WH_l , $WH_{l'}$, let $c = \|WH_l - WH_{l'}\|_p$; they are hashed to the same value with probability $P(c)$:

$$\begin{aligned}
 P(c) &= \Pr_{a,b} [h_{a,b}(WH_l) = h_{a,b}(WH_{l'})] \\
 &= \int_0^r \frac{1}{c} f_p\left(\frac{t}{c}\right) \left(1 - \frac{t}{r}\right) dt.
 \end{aligned} \tag{11}$$

Then spatial and temporal mapping rules are established for ST to expand the number of mapping blocks of ST. Here four mapping rules are established, shown as Rules 1–4. The first three mapping rules take advantage of a coherency sensitive hashing technique [14] to mine spatial correlations between frames, and Rule 4 makes direct use of temporal correlations of video sequences.

Rule 1. If $g_m(b_k^n) = g_m(b_l^{n+1})$, then $b_k^n \mapsto b_l^{n+1}$, where $b_k^n \in B^n$, $b_l^{n+1} \in B^{n+1}$.

Rule 2. If $\text{left}(b_k^n) \mapsto b_l^{n+1}$, then $b_k^n \mapsto b_l^{n+1}$, where $b_k^n \in B^n$, $b_l^{n+1} \in B^{n+1}$, and $\text{left}(b_k^n)$ is the left neighboring block of b_k^n .

FIGURE 3: Tree scheme for computing projections of a 2D signal onto WH kernels of order 8. In the Walsh-Hadamard kernel array, yellow denotes the value 1 and blue denotes the value -1 .

FIGURE 4: Comparison of interpolation frames obtained using six methods. From top to bottom, every row shows different interpolation results: interpolation frame 44 of walk, interpolation frame 31 of jump in place, interpolation frame 205 of silent, and interpolation frame 3 of space. From left to right, different algorithms are used: TSS, ARPS, H&S, Classic+NL-Full, CSH, and RGFI.

FIGURE 5: Comparison of PSNR using six algorithms for walk, jump in place, silent, and space, respectively.

Rule 3. If $g_m(b_k^n) = g_m(b_{k'}^n)$ and $b_{k'}^n \mapsto b_l^{n+1}$, then $b_k^n \mapsto b_l^{n+1}$, where $b_k^n \in B^n$, $b_{k'}^n \in B^n$, $b_l^{n+1} \in B^{n+1}$.

Rule 4. If the coordinate values of the upper left corner of b_k^n are equal to those of b_l^{n+1} , then $b_k^n \mapsto b_l^{n+1}$, where $b_k^n \in B^n$, $b_l^{n+1} \in B^{n+1}$.

The third step is to determine the nearest blocks from the obtained mapping blocks of ST. Here a freedom search technique [8] using the patchmatch method is used to initialize the nearest blocks for each ST as follows:

$$b_{\text{near}}^{n+1} = b_k^n + w\alpha^i R^i. \quad (12)$$

In (12), $R^i \in [-1, 1] \times [-1, 1]$, α is a constant value, w is a search radius, and b_{near}^{n+1} is the best block of any ST b_k^n . Then for each hash table, (13) is used to compare and update the nearest blocks:

$$b_{\text{near}}^{n+1} = b_{\text{map}}^{n+1}, \quad \text{if } \text{disc}(b_k^n, b_{\text{map}}^{n+1}) < \text{disc}(b_k^n, b_{\text{map}}^n). \quad (13)$$

Finally, the predicted value of the pixels in the ST can be computed through a smoothing operation for the nearest blocks obtained as follows:

$$I'(x_i, y_i) = \frac{\left(\sum_{j=1}^{|A^n(b_k^n)|} I(x_j^{n+1}, y_j^{n+1}) \right)}{|A^n(b_k^n)|}, \quad (14)$$

where $I'(x_i, y_i)$ denotes the predicted value of an interpolated pixel, $A^n(b_k^n)$ is the nearest blocks set of b_k^n , $I(x_j^{n+1}, y_j^{n+1}) \in b_j^{n+1}$, and $b_j^{n+1} \in A^n(b_k^n)$. In addition, to obtain color frame interpolation, the R, G, and B channel values of each pixel can be computed in the same way.

4. Experimental Results and Discussion

In this section, the proposed RGFI algorithm is compared with other frame interpolation algorithms, including the block matching methods of three-step search (TSS), the adaptive rood pattern search (ARPS) [23], the optical flow method

FIGURE 6: Comparison of RMSE using six algorithms for walk, jump in place, silent, and space, respectively.

of Horn and Schunck (H&S) [24], Classic+NL-Full [5], and CSH [14]. Each algorithm was run in MATLAB using a PC with Intel(R) Pentium(R) 4 CPU processor, 3.00 GHz, and 3 GB of main memory. To evaluate algorithm performance, one in every two frames of the original video sequences was removed. Then this removed frame was reconstructed using different frame interpolation algorithms, and the reconstructed frame was compared with the removed frame. In all experiments, the detailed parameters were as follows. (1) Since c affects the accuracy of motion region detection, we try several different values and finally determine its value to be 0.2. (2) According to [11], we found that the best choice of parameters $\{L, \phi_1, \phi_2, \phi_3\}$ is $\{17, 3, 10, 20\}$, respectively, and this provides good discrimination ability. (3) Adjustment parameters $\{\varepsilon_L, \varepsilon_R, \varepsilon_T, \varepsilon_B\}$ are used to complete the conversion from key points to motion regions and by experiments we found that the best choice of each parameter is 8. (4) The value of α is 0.5 and this is the same with [8]. (5) Based on [14], we choose three different values for s , namely, 8, 16, and 32. At the same time, we find that the appropriate value is 16 according to the obtained experiment results. The test video sequences used for these experiments were walk (640×480), jump in place (320×240), silent (176×144), and space (320×240). Walk and jump in place were provided by the image sequence

evaluation research laboratory in Barcelona [25]. Silent was obtained from the video trace research group at Arizona State University [26]. Space was obtained from Youku Web. The selection of test video sequences covered a variety of different background and object motions, which are frequently found in real video.

Figure 4 shows the frame interpolation results using the six algorithms. Every row in the figure shows the interpolated frames for the same video using different algorithms, and every column shows the interpolation results for different videos using the same algorithm. From the red-bordered region of each figure, the visual differences among algorithms can be determined. TSS exhibited a poor interpolation effect, ARPS could not reveal the whole motion content, H&S and Classic+NL-Full introduced a suspension effect, and CSH produced disappointing effects, such as vague legs on the walking man. From these figures, it is evident that the proposed algorithm shows comparatively better performance in terms of visual quality.

To validate the algorithm further, objective measurements are also provided. Peak signal-to-noise ratio (PSNR) and root mean square error (RMSE) are traditional quantitative measures of accuracy. Figures 5 and 6 show their values for the interpolation frames of different video sequences

FIGURE 7: Comparison of MSSIM using six algorithms for walk, jump in place, silent, and space, respectively.

using different algorithms. In these figures, the red curves represent the proposed algorithm, and the five black curves using a different format represent comparison algorithms. It can be observed that the PSNR values for the interpolation frames obtained using the proposed algorithm are generally the highest, and the RMSE values are the lowest. However, the proposed method might produce unsatisfactory results, for example, the PSNR value of interpolation frame 58 of walk. Overall, it can be clearly demonstrated that the proposed new method outperforms the other five algorithms.

The MSSIM measure was also used to evaluate the visibility quality of interpolation frames. MSSIM assesses the image visibility quality from an image formation point of view under the assumption of a correlation between human visual perception and image structural information [27, 28]. Figure 7 shows MSSIM comparison results using the six frame interpolation algorithms for the different video sequences. Note that the proposed RGFI algorithm generally achieved a greater MSSIM than the other five algorithms, and the results show that the interpolation frames obtained using

the proposed algorithm are closer to the original frames in terms of image structure similarity.

Table 1 summarizes the average PSNR, RMSE, and MSSIM values for each video sequence using different algorithms. From this table, it can be significantly determined that RGFI always obtains the highest PSNR and MSSIM and the lowest RMSE. In a word, because of its encouraging performance in terms of video visualization and quantitative quality assessment, the proposed algorithm is very competitive in frame interpolation.

5. Another Contribution

We combine RGFI with a seam carving approach [29] to achieve video resizing, so as to obtain high-quality resizing results displayed on the monitoring display devices with different resolutions. Figure 8 shows comparative results for frame 10 of the video of “walk with dog” using four methods: scaling method with uniform resizing ratio, the best cropping method directly using cutting, the seam carving method by removing or duplicating seams, and the proposed algorithm

FIGURE 8: Comparative results for a walking man with a dog for the four resizing methods when the resolutions are resized from 180×144 to 200×288 .

using accurate motion regions obtained from RGFI. It can be seen that using the scaling method (see Figure 8(b)), the walking man and his dog all become vaguer than before. Using the best cropping (see Figure 8(c)), the walking man is only partly displayed, resulting in missing original information. Only using seam carving (see Figure 8(d)) does the prominent part of this video sequence change much less than before. From Figure 8(e), it is apparent that the proposed method can protect the prominent object of the original frame when the video sequence resolutions are changed. Table 2 summarizes five evaluation indicators, including average gradient (AG), information entropy (IE), edge intensity (EI), spatial frequency (SF), and image definition (ID) values. These indicators are used to measure the resizing quality of the video frames. From this table, it is clear that the proposed method achieved the highest AG, IE, EI, SF, and ID values, which indicates that the proposed method can effectively improve resizing quality when image sequence resolutions are changed.

6. Conclusions

In this paper, promising RGFI method for intelligent monitoring systems has been presented. The main feature of this method is its ability to obtain relatively high-quality of interpolated frames according to spatial and temporal correlations in video sequences. The implementation process involves two steps: motion region detection and interpolated pixel computation. The former determines the prediction range of interpolation frames through a detection approach based on visual correspondence, and the latter computes interpolated pixels using spatial and temporal mapping rules based on coherency sensitive hashing. Experimental results show that the proposed algorithm outperforms the other five representative frame interpolation algorithms examined on subjective quality and in quantitative measures. At the same time, RGFI combined with a seam-carving approach can achieve video resizing. In a word, a promising frame interpolation method has been proposed for the intelligent

TABLE 1: Comparison of average PSNR, RMSE, and MSSIM values.

Method	Evaluation	Walk	Jump-in-place	Silent	Space
TSS	PSNR	33.6385	37.1343	31.8166	34.1798
	RMSE	15.8233	11.4544	21.0495	15.3359
	MSSIM	0.9351	0.9402	0.9065	0.9647
ARPS	PSNR	34.2260	37.3584	32.0574	33.1645
	RMSE	15.8436	11.08734	20.4844	16.7526
	MSSIM	0.9393	0.9405	0.9125	0.9616
H&S	PSNR	27.7678	34.7418	29.3596	25.3645
	RMSE	31.5615	15.5741	29.3785	40.4711
	MSSIM	0.8989	0.9296	0.8757	0.8698
Classic+NL-Full	PSNR	28.3820	35.9997	29.9458	26.3099
	RMSE	30.4592	14.1522	27.1623	36.5059
	MSSIM	0.9210	0.9456	0.8953	0.8761
CSH	PSNR	29.2065	32.4474	32.9375	22.0820
	RMSE	26.4691	21.8114	18.8703	53.6209
	MSSIM	0.9325	0.9351	0.9275	0.9141
Proposed algorithm	PSNR	35.4886	38.4762	33.4151	34.8564
	RMSE	12.7313	9.7977	17.9540	14.3546
	MSSIM	0.9472	0.9518	0.9323	0.9717

TABLE 2: Comparative results for the four resizing methods.

Method	AG	IE	EI	SF	ID
Scaling	10.8870	14.1536	106.7085	7.6176	27.9762
Best cropping	8.5673	9.6485	88.1226	7.5842	20.9809
Seam carving	9.6769	13.5044	96.8119	7.5244	26.1953
Proposed method	11.6518	14.3028	116.5497	7.6478	28.1341

monitoring systems. However, as a new frame interpolation algorithm, RGFI also has its disadvantage and the disadvantage is to take comparatively longer time caused by higher complexity. In the future, we will exploit a multicore architecture to do parallel computing, so as to reduce running time of the algorithm.

Acknowledgments

This work was supported by the National Basic Research Program of China (973 Program) 2012CB821200 (2012CB821206), the National Natural Science Foundation of China (nos. 91024001, and 61070142), and the Beijing Natural Science Foundation (no. 4111002).

References

- [1] T. Stich, C. Linz, C. Wallraven, D. Cunningham, and M. Magnor, "Perception-motivated interpolation of image sequences," *ACM Transactions on Applied Perception*, vol. 8, no. 2, pp. 1–25, 2011.
- [2] K. Chen and D. A. Lorenz, "Image sequence interpolation using optimal control," *Journal of Mathematical Imaging and Vision*, vol. 41, no. 3, pp. 222–238, 2011.
- [3] T. Brox and J. Malik, "Large displacement optical flow: descriptor matching in variational motion estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 3, pp. 500–513, 2011.
- [4] M. Carlini, S. Castellucci, M. Guerrieri, and T. Honorati, "Stability and control for energy production parametric dependence," *Mathematical Problems in Engineering*, vol. 2010, Article ID 842380, 21 pages, 2010.
- [5] D. Sun, S. Roth, and M. J. Black, "Secrets of optical flow estimation and their principles," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 2432–2439, June 2010.
- [6] Y.-W. Tai, S. Liu, M. S. Brown, and S. Lin, "Super resolution using edge prior and single image detail synthesis," in *Proceedings of the 23rd IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 2400–2407, June 2010.
- [7] Z. Shi, W. A. C. Fernando, and A. Kondo, "Adaptive direction search algorithms based on motion correlation for block motion estimation," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 3, pp. 1354–1361, 2011.
- [8] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "PatchMatch: a randomized correspondence algorithm for structural image editing," *ACM Transactions on Graphics*, vol. 28, no. 3, article 24, 2009.
- [9] J. Yang, Y. F. Li, K. Wang, Y. Wu, G. Altieri, and M. Scalia, "Mixed signature: an invariant descriptor for 3D motion trajectory perception and recognition," *Mathematical Problems in Engineering*, vol. 2012, Article ID 613939, 29 pages, 2012.
- [10] S. Cong and Z.-B. Sheng, "On exponential stability conditions of descriptor systems with time-varying delay," *Journal of Applied Mathematics*, vol. 2012, Article ID 532912, 12 pages, 2012.
- [11] M. Ambai and Y. Yoshida, "CARD: Compact and real-time descriptors," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, pp. 97–104, November 2011.

- [12] C. Huo, C. Pan, L. Huo, and Z. Zhou, "Multilevel SIFT matching for large-size VHR image registration," *IEEE Geoscience and Remote Sensing Letters*, vol. 9, no. 2, pp. 171–175, 2012.
- [13] J. Kybic and I. Vnučko, "Approximate all nearest neighbor search for high dimensional entropy estimation for image registration," *Signal Processing*, vol. 92, no. 5, pp. 1302–1316, 2012.
- [14] S. Korman and S. Avidan, "Coherency sensitive hashing," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, pp. 1607–1614, November 2011.
- [15] H. Yin, Y. Chai, S. X. Yang, and X. Yang, "Fast-moving target tracking based on mean shift and frame-difference methods," *Journal of Systems Engineering and Electronics*, vol. 22, no. 4, pp. 587–592, 2011.
- [16] Z. Yan, D. Xu, and M. Tan, "A fast and robust method for line detection based on image pyramid and Hough transform," *Transactions of the Institute of Measurement and Control*, vol. 33, no. 8, pp. 971–984, 2011.
- [17] P. Mainali, Q. Yang, G. Lafruit, L. Van Gool, and R. Lauwereins, "Robust low complexity corner detector," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 4, pp. 435–445, 2011.
- [18] V. Javier Traver and A. Bernardino, "A review of log-polar imaging for visual perception in robotics," *Robotics and Autonomous Systems*, vol. 58, no. 4, pp. 378–398, 2010.
- [19] E. Brun, A. Guittet, and F. Gibou, "A local level-set method using a hash table data structure," *Journal of Computational Physics*, vol. 231, no. 6, pp. 2528–2536, 2012.
- [20] M. T. Hamood and S. Boussakta, "Fast Walsh-Hadamard-Fourier transform algorithm," *IEEE Transactions on Signal Processing*, vol. 59, no. 11, pp. 5627–5631, 2011.
- [21] G. Ben-Artzi, H. Hel-Or, and Y. Hel-Or, "The gray-code filter kernels," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 3, pp. 382–393, 2007.
- [22] Y. Hel-Or and H. Hel-Or, "Real-time pattern matching using projection kernels," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 9, pp. 1430–1445, 2005.
- [23] A. Barjatya and Y. Yoshida, "Block matching algorithms for motion estimation," DIP 6620 Spring 2004 Final Project Paper, 2004.
- [24] S. N. Tamgade and V. R. Bora, "Motion vector estimation of video image by pyramidal implementation of Lucas Kanade Optical flow," in *Proceedings of the 2nd International Conference on Emerging Trends in Engineering and Technology (ICETET '09)*, pp. 914–917, December 2009.
- [25] Research lab on image sequence evaluation, <http://iselab.cvc.uab.es/files/Tools/CvcActionDataSet/index.htm>.
- [26] YUV video Sequences, <http://trace.eas.asu.edu/yuv/index.html>.
- [27] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [28] L. Zi and J. Du, "Energy-driven image interpolation using Gaussian process regression," *Journal of Applied Mathematics*, vol. 2012, Article ID 435924, 13 pages, 2012.
- [29] S. Avidan and A. Shamir, "Seam carving for content-aware image resizing," *ACM Transactions on Graphics*, vol. 26, no. 3, Article ID 1276390, 2007.