Expert Systems with Applications 42 (2015) 751–759

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Color video denoising using epitome and sparse coding

Hwea Yee Lee, Wai Lam Hoo, Chee Seng Chan *

Centre of Image and Signal Processing, Faculty of Computer Science & Information Technology, University of Malaya, 50603 Kuala Lumpur, Malaysia

ARTICLE INFO

Article history:
Available online 8 September 2014

Keywords:
Video denoising
Video epitome
Sparse coding

ABSTRACT

Denoising is a process that remove noise from a signal. In this paper, we present a unified framework to deal with video denoising problems by adopting a two-steps process, namely the video epitome and sparse coding. First, the video epitome will summarize the video contents and remove the redundancy information to generate a single compact representation to describe the video content. Second, employing the single compact representation as an input, the sparse coding will generate a visual dictionary for the video sequence by estimating the most representative basis elements. The fusion of these two methods have resulted an enhanced, compact representation for the denoising task. Experiments on the publicly available datasets have shown the effectiveness of our proposed system in comparison to the state-of-the-art algorithms in the video denoising task.

1. Introduction

Since decades, research in denoising has grown progressively into one of the important research in the image processing domain. One of the main reasons is signals, including audios, images, and video sequences, are subjected to random noise contamination during the process of signals acquisition or transmitting. Besides that, low-end imaging devices such as mobile phones and digital cameras have became ubiquitous, there are ever more demands for reliable denoising solutions as a good denoising solution will be able enhance the performance of subsequent processes such as compression (Papadopoulos, Kalogeiton, Chatzichristofis, & Papamarkos, 2013), segmentation (Goncalves & Bruno, 2013), recognition (Chan & Liu, 2009), object detection (Karasulu & Korukoglu, 2012) and tracking (Lim, Tang, & Chan, 2014b).

This paper is primarily focused on video denoising, and therefore only related work in this area will be reported. Interested readers are encouraged to refer to Buades, Coll, and Morel (2005), Milanfar (2013) and Rao and Chen (2012) for a comprehensive review on the related literature. Principally, video denoising can be classified into two broad categories: the spatial domain and the transform domain, respectively. In the former, pixel information is utilized to perform the denoising, while in latter, the spatial frequency spectrum is employed. From here, it can be further detailed into methods that either employ spatio-temporal or temporal correlation approaches. The spatio-temporal approaches use both the spatial and temporal information in denoising, but did not compensate with motion features. On the other hand, the temporal correlation approaches are performed based on motion compensation filters. These filters are employed to create estimated trajectories to remove the temporal non-stationarity of the video for the denoising task. In this work, we intend to address the video denoising in spatial domain using the spatial–temporal approaches. Particularly, we extended the Benoît, Mairal, Bach, and Ponce (2011) work from 2D to 3D, i.e. from image domain to video domain utilize both the sparse coding and video epitome in a unified framework, namely the video epitome and sparse coding framework (VESC), as well as from monocular image to color video.

Sparse coding consists in representing the signals (data vectors) as sparse linear combinations of basis elements, and has been shown to achieve better performance on denoising. For example, Peyré (2009) proposed a generative model for textures using the sparse description of image content. This model is based on a sparse expansion of texture patches into a redundant dictionary. To synthesize a texture, an optimization solution to find the texture that having sparse patches in the dictionary is performed. While most of the proposed sparse representations for signals were used to deal with monocular images, Mairal, Elad, and Sapiro (2008) investigated on learning the dictionary using color image. Instead of creating individual dictionaries for each color channel and perform the denoising separately on each color channel in the noisy image, they concatenated the RGB values to a vector and trained on those directly. Empirically, this method has proofed to be more effective in color image denoising task compare to model each of the channel separately. However, accordingly to Benoît et al., 2011, the sparse coding in this paper extracts image

* Corresponding author. Tel.: +60 379676433.
E-mail addresses: leehyde@siswa.um.edu.my (H.Y. Lee), wailam88@siswa.um.edu.my (W.L. Hoo), cs.chan@um.edu.my (C.S. Chan).

http://dx.doi.org/10.1016/j.eswa.2014.08.033
0957-4174/

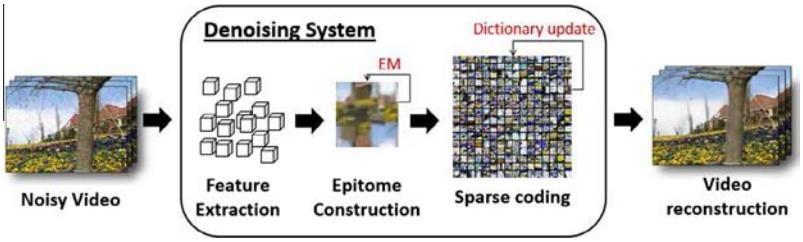


Fig. 1. A summary of our denoising system. From the noisy video, we extract the 3D video cube features. Then, we learn the video epitome from these 3D features, through Expectation–maximization (EM) algorithm. The dictionary is learnt using the sparse coding from the converged epitome, and is updated iteratively. Finally, we reconstruct the video via the converged sparse dictionary.

patches information from unstructured set of patches in a random manner, and such solution lacks of shift-invariance properties.

Therefore, Benoît et al. (2011) and Aharon and Elad (2008) proposed a variant – an epitome-based dictionary formulation to handle this limitation. The work unify both the epitome approach (Jojic, Frey, & Kannan, 2003) and dictionary learning framework by allowing an image patch to be represented as a sparse linear combination of several patches extracted from the epitome. Epitome is a simple appearance and shape model for image. More generally, it is a summarized version of the image which retains the visual quality as the visual input. As such, Benoît et al. (2011) and Aharon and Elad (2008) differed from the aforementioned dictionary learning with the introduction of structured dictionaries which can be obtained from the epitome. Though it had shown promising denoising results, these work are currently constrained to monocular image only. Our proposed method extend Benoît et al. (2011) in such a way that our video denoising approach require only one video epitome and one KSVD dictionary, while Benoît et al. (2011) requires one epitome and one KSVD dictionary for each frame in a video clip. With this, our proposed approach greatly reduce the computational cost; while in the meantime, empirically, obtain good denoising results compare to Benoît et al. (2011) in publicly available datasets.

As a summary, our main contribution is we extended the Benoît et al. (2011) work (1) from image domain to video domain, and (2) from monocular image to color image sequences. Our motivation is nowadays, the resolution of display devices is often much higher than that of video, particularly in the case of video streamed over the internet, and a solution that is being able to exploit the high resolution of modern display devices when rendering video is essential. Nonetheless, as aforementioned, a good denoising solution will be able to enhance the performance of subsequent processes. Experimentally, we first show that our proposed system is comparable to the state-of-the-arts approaches in removing additive white Gaussian noise (AWGN) on benchmark videos. Then, we show the importance of denoising as a preprocessing step for further analysis of video sequences (i.e. object tracking).

The rest of the paper is structured as follows: we will discuss the related work in Section 2. Then, we will explain how our denoising system work, and how it differs from Benoît et al. (2011) in Section 3. Following on, we show the performance of the proposed framework in terms of publicly available datasets, and compare with state-of-the-arts approaches in Section 4. Besides, we also demonstrate the importance of a good denoising algorithm in object tracking application. Finally, we conclude our findings in Section 5.

2. Related work

Existing denoising methods can be categorized into spatial and transform domain, respectively where the spatial domain (Aharon & Elad, 2008; Benoît et al., 2011; Cheung, Frey, & Jojic, 2008; Elad & Aharon, 2006; Jojic et al., 2003; Mairal et al., 2008; Peyré, 2009; Protter & Elad, 2009) utilizes pixel information to denoise, while the transform domain (Blu & Luisier, 2007; Dabov, Foi, & Egiazarian, 2007; Dai, Au, Pang, & Zou, 2013; Dai et al., 2010; Eksioglu, 2014; Varghese & Wang, 2010; Wang, Yang, & Fu, 2010; Wu, Cao, Tao, & Zhuang, 2013; Yang & Ren, 2011) make use of spatial frequency spectrum to reduce the noise. Some of these research works focus on spatial–temporal approaches without the motion compensation cues (Boulanger et al., 2010; Dabov et al., 2007; Kuang, Zhang, & Yi, 2014; Protter & Elad, 2009; Rubinstein, Zibulevsky, & Elad, 2010), while the rest utilize the motion compensation filters (Wang et al., 2010; Yang & Ren, 2011). This paper is primarily focused on video denoising, and therefore only the related work in this area will be reported. Readers can refer to Buades et al. (2005), Milanfar (2013) and Rao and Chen (2012) for more comprehensive survey.

Dictionary learning has been actively used in machine learning domain for image classification problems, such as the Bag-of-Words model and spatial pyramid matching (Lazebnik, Schmid, & Ponce, 2006). In recent years, sparse dictionary learning has been very popular as its ability to enhance the shift-invariance properties via direct sparse decomposition technique over redundant dictionaries. In the denoising domain, few notable works are also found to utilize this concept. For instance, Elad and Aharon (2006) learn the sparse dictionary using KSVD algorithm for image denoising, and Protter and Elad (2009) extends Elad and Aharon (2006) to the video domain. However, the denoising results from all these methods encounter a fluctuation issue due to the randomness of the image patches, i.e. these methods learn the sparse dictionary using randomly selected image patches from the noisy image.

Therefore, Aharon and Elad (2008) introduce the Image-Signature-Dictionary (ISD) that is similar to the epitome approach (Jojic et al., 2003), but replaces the epitome initialization with sparse representations. With this, ISD approach uses less memory and obtains shift and scale flexibilities. Apart from the ISD, Benoît et al. (2011) proposed a different strategy, i.e using the epitome approach similar to Jojic et al. (2003) that apply epitome initialization to form a compact representation for the noisy image. From the generated epitome, they extract all the overlapping patches to learn the sparse dictionary. Both methods aim to obtain the best image patches to learn the dictionary. Though the denoising results are promising, both methods are currently limited to monocular image denoising task only.

Therefore, in this paper, we extend Benoît et al. (2011) from monocular image domain to color video domain. We employ the epitome to find those important patches from noisy video, and learn the sparse dictionary from the built epitome. As indicated in Benoît et al. (2011), this unified framework would gain enhanced shift-invariance properties. Particularly, the proposed method is benefiting from both the advantages of epitome that is built using overlapping patches in compact representation, and sparse dictionary that have sparse and redundant representation.

(a) Init. (b) Visualization of the EM algorithm from 1-10 iteration

Fig. 2. First row: Example frames from the 'leaves' video. Second row: Example epitome visualized from each EM iterations. From left to right: 1st to 10th iterations.

3. Methodology

In this section, we discuss the proposed VESC framework in detailed. From a noisy video, we first extract video cubes from the video as 3D features. Then, we learn the epitome from these features in a condensed volume representation. Subsequently, we learn the sparse dictionary from the built epitome, i.e. in our approach, the atoms for the dictionary are those patches extracted from the video epitome, which is different from those traditional methods that directly sample patches from the input video. Finally, we reconstruct the video using the converged sparse dictionary. Fig. 1 shows the summary of our framework.

3.1. Feature extraction and epitome construction

To extract the features, we adopt the feature extraction method as to Cheung et al. (2008). Given a video sequences V, a set of 3D video cubes $\{Z_k\}_{k=1}^{K}$ are extracted randomly. The 3D cubes can be in any size, but in our paper, we use cube size of $F \times N \times N$, where F is the number of frames, and N represents the patch size. These 3D cubes contain an ordered set of pixels indexed by their coordinates in the video: $z_k = \{z_{\mathcal{I},k}\},\ \mathcal{I} \in S_k = \{(x,y,t),(x+1,y+1,t+1),\ldots,(x+N,y+N,t+F)\}$, where $\mathcal{I}$ is (x,y,t) coordinate. These coordinates will be stored in an ordered set S_k.

A video epitome E of size $p \times q \times r$ is a condensed version of the corresponding input video V of size $\mathcal{X} \times \mathcal{Y} \times T$ where $p \ll \mathcal{X},\ q \ll \mathcal{Y}$ and r is the number of frames a video epitome will be consisted (Cheung et al., 2008). Let $Z = \{Z_k\}_{k=1}^{K}$ be the patch level representation of V, i.e., is the set of all possible patches from V. The video epitome E corresponds to V is estimated using Z and represents the salient visual contents of V effectively. More specifically, the video epitome E is derived by searching for a set of patches in E that corresponds to the set Z based on Gaussian probability distribution. The patches in E are defined by a set of mapping, $T = T_k(\mathcal{I})$, which shows a displacement between two patches V and E respectively. Assuming distribution at each video epitome location to be Gaussian, the conditional probability for mapping patches in epitome to set of patches in a video should fit the following generative model:

$$p(Z_k|T_k,E) = \prod_{i \in S_k} \mathcal{N}(z_{\mathcal{I},k}; \mu_{T_k}(\mathcal{I}), \phi_{T_k}(\mathcal{I})) \tag{1}$$

where $\mu_{T_k}(\mathcal{I}),\ \phi_{T_k}(\mathcal{I})$ are the mean and variance of a Gaussian distribution $\mathcal{N}$, T is the mapping function, and k represents the number of cubes extracted.

Solving the maximum likelihood problem leads to expectation maximization algorithm. We initialize μ to be normally distributed using the mean pixel intensity of the video (Fig. 2a). During the E-step, optimum T_k from Z_k to E is estimated. For each of the input patch k, we will find the posterior distribution as:

$$p(T_k|Z_k,E) = \frac{p(Z_k|T_k,E)\rho(T_k)}{\sum_{k \in K} p(Z_k|T_k,E)\rho(T_k)} \tag{2}$$

For the M-step, given the new set of T_k, the μ and ϕ at epitome's coordinate (e_x,e_y,e_t) are computed, where $\rho(T_k)$ is the mean of all the pixel intensities from the video patches:

$$\mu_{(e_x,e_y,e_t)} = \frac{\sum_{\mathcal{I}}\sum_k [(e_x,e_y,e_t) = T_k(i)] z_{\mathcal{I},k}}{\sum_{\mathcal{I}}\sum_k [(e_x,e_y,e_t) = T_k(\mathcal{I})]} \tag{3}$$

$$\phi_{(e_x,e_y,e_t)} = \frac{\sum_{\mathcal{I}}\sum_k [(e_x,e_y,e_t) = T_k(\mathcal{I})] (z_{\mathcal{I},k} - \mu_{(e_x,e_y,e_t)})^2}{\sum_{\mathcal{I}}\sum_k [(e_x,e_y,e_t) = T_k(\mathcal{I})]} \tag{4}$$

This process will be terminated once the convergence is reached. We summarize the epitome construction process in Algorithm 1. Fig. 2 shows the original video sequence ('leaves' sequence) and the corresponding transition of the epitome from iteration 1st to 10th.

Algorithm 1. Epitome construction

Require: Randomly initialized video cubes $\{Z_k\}_{k=1}^{K}$
Ensure: All parameters are set: epitome size, number of iterations
1. Initialize epitome, $E = \{\mu, \phi\}$ by normally distributed video mean pixel intensity.
repeat
a. E-Step: Fix mean, μ and variance, ϕ to optimize target function T_k.
b. M-Step: Fix target function T_k to optimize μ and ϕ.
until maximum number of iterations are reached.

3.2. Sparse coding

In this section, we will discuss how the constructed epitome is unified in the proposed framework to learn the sparse dictionary. A summary of the approach is shown in Algorithm 2.

3.2.1. Dictionary learning

The idea of learning dictionaries was first proposed by Olshausen and Field (1997) in 1996, aiming at given a set of $Y = (\gamma_j)_{j=1}^{m} \in \mathbb{R}^{n \times m}$ of m signals $\gamma_j \in \mathbb{R}^m$, find the best dictionary $D = (d_i)_{i=1}^{p}$ of p atoms $d_i \in \mathbb{R}^n$ to all the data. More generally, the dictionary learning perform an optimization both on the dictionary

(a) $\sigma = 10$ (b) $\sigma = 30$ (c) $\sigma = 50$

Fig. 3. Example of KSVD codebook on different noise levels using the 'leaves' video. From left to right: $\sigma = 10,\ 30,\ 50$.

D and a set of coefficients $\alpha' = (\alpha_j)_{j=1}^{m} \in \mathbb{R}^{p\times m}$ where for $j = 1,\ldots,m, \alpha_j$ is the set of coefficients of the data γ_j. In this paper, each $\gamma_j \in \mathbb{R}^n$ is a patch of size $n = N \times N$ extracted from the video epitome, E (Section 3.1). We consider a dictionary $D \in \mathbb{R}^{n\times p}$ of $p \geqslant n$ atoms in $\mathbb{R}^n$. The initial dictionary D is computed by a random selection of patches from E, and we normalised them to be an unit-norm. Then, fixing the D and let the video epitome E as the current residuals, we optimize the α_{ij} according to:

$$\forall_{ij}\, \alpha_{ij} = \min_{\alpha} \|D\alpha_{ij} - E\|_2^2 \tag{5}$$

That is, the mechanism starts with finding an atom in the dictionary that can show the maximum correlation with the current residuals. In other words, we first find the best atom in the dictionary D that can minimize the error, $Err = \|D\alpha - E\|_2^2$ between the columns, D. If Err does not satisfy the target threshold $\sqrt{p}C\sigma$, where p is total number of atoms d, C is a noise gain of 1.15 and σ is the noise applied to the video, another atom will be selected from D to further minimize Err. These steps are repeated until the Err satisfies the target threshold. We summarize these steps in Algorithm 2. Note that, the optimization algorithm was adopted as a part of the KSVD algorithm Elad and Aharon (2006). Fig. 3 shows sample dictionary generated based on the 'leaves' video in different noise levels.

Algorithm 2. Sparse coding

Require: Constructed video epitome from Algorithm 1
Ensure: All parameters are set: number of atoms, maximum number of iterations, atom optimization threshold
repeat
1. Search sequentially an atom, d in the dictionary that can minimize the error Err between the columns of D with the current residuals,
2. If Err does not satisfy the error threshold, new atom, d' is added into set of selected atoms, d_{set}
3. Update the residuals by projecting the video epitome E onto the d_{set}.
until $\|D\alpha - E\|_2^2 < \sqrt{p}C\sigma$.

3.2.2. Dictionary update and video reconstruction

Once the sparse coefficients α_{ij} are computed, one can update the dictionary. Let d_l where $l \in 1\ldots L$ represents the columns in dictionary D, we select patches that are associated with the atom, $\omega_l = \{(i,j)$ such that $\alpha_{ij}(l) \neq 0\}$. Then, we compute the representation error for each index $(i,j) \in \omega_l$ according to

$$e_{ij}^{l} = R_{ij}E - D\alpha_{ij} + d_l\alpha_{ij}(l) \tag{6}$$

where R_{ij} is the operator that extracts patches from E. We set G_l as the matrix whose columns are the e_{ij}^{l}, and α^{l} as the row vector whose elements are the $\alpha_{ij}(l)$. Finally, we update d_l and the $\alpha_{ij}(l)$ by minimizing:

$$\arg\min_{\alpha_l, d_l} \quad \|G_l - d_l\alpha_l\|_2^2 \tag{7}$$

The dictionary update is considered accomplished once all the atoms in D are updated; and the VESC learning will be terminated once the maximum number of iterations are reached. The denoised video will be reconstructed using the converged sets of sparse coefficients $\hat{\alpha}$:

$$\hat{V} = D\hat{\alpha} \tag{8}$$

where $\hat{V}$ is the denoised video sequence.

4. Experiments

In this section, we show the effectiveness of the proposed method in publicly available datasets, and a comparison with the state-of-the-art solutions. Unless specified, our proposed method will use these settings for all experiments. The epitome size is $p = q = 40$, and $r = 1$. The number of frames that used in the feature extraction, F, is 2. and the number of iterations for epitome is set to 10. For the sparse coding, the dictionary has 256 atoms, and it is set to use 20 iterations during the denoising process. For the video contamination process, we apply AWGN in multiple noise levels for different experiment purposes.

To make a video sequences clearer or subjectively better, we tend to have different criteria to evaluate with. In here, we provide visualization in different experimental settings, as well as a PSNR evaluation (Eq. (9)) as a standard benchmark to compare with the state-of-the-art solutions.

$$\text{PSNR} = log_{10}\frac{65025}{3\mathcal{Z}^{-1}\sum_{c=R,G,B}\sum_{\mathcal{I}}(V_c(\mathcal{I}) - \hat{V}_c(\mathcal{I}))^2} \tag{9}$$

where $\mathcal{I}$ denotes the spatial coordinate of pixels (as in Section 3.1) and $\mathcal{Z}$ is the total number of pixels in an image.

Patch size selection: To select the optimum patch size for our denoising system, we first run experiments on the 'leaves' video using different patch sizes. The 'leaves' video is publicly available (Cheung et al., 2008) and contains 17 frames with a frame size 200 × 150. We perform experiments using $N = \{6, 8, 10, 12\}$ and the results are shown in Fig. 4. It is clear that $N = 8$ is the optimum settings for the denoising task as it has the highest PSNR. Besides, we visualize the 'leaves' frames in Fig. 5 to show that $N = 8$ gives optimum result qualitatively. From the Fig. 5, $N = \{6, 10, 12\}$ seem to lose more background details than $N = 8$. Based on this, we had chosen $N = 8$ throughout the rest of our experiments.

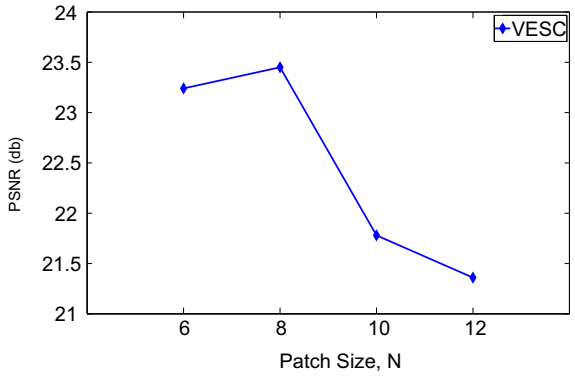


Fig. 4. The PSNR(db) results in terms of different patch sizes, N using the 'leaves' video.

Comparison with related methods: Since our denoising system are built based on the idea of Benoît et al. (2011), Cheung et al. (2008) and Elad and Aharon (2006), therefore we compare our denoising performance with all these three approaches to show the effectiveness of the proposed method. Note that, since Benoît et al. (2011) and Elad and Aharon (2006) are image denoising framework, hence frame-by-frame denoising is performed for the video denoising task.

In this experiment, we employed the 'leaves' and 'tennis' videos. The 'tennis' video consists of a size of 240 $\times$ 352, with 150 frames. Fig. 6 presents the sub-sample denoising performance for Benoît et al. (2011), Cheung et al. (2008), Elad and Aharon (2006) and VESC qualitatively. Visually, we could notice that resultants from Cheung et al. (2008) smoothen both the videos, and therefore have a relatively low PSNR as illustrated in Fig. 6a and b, respectively. On the other hand, Benoît et al. (2011), Elad and Aharon (2006) and VESC have very competitive results, and hard to be differentiated. To handle this issue, we deduce another comparison in Fig. 7b and d for both the 'leaves' and 'tennis' video to highlight the advantages of VESC over Elad and Aharon (2006) and Benoît et al. (2011). In most of the frames for both videos, VESC are shown to outperform the Elad and Aharon (2006) in frame-to-frame basis by a clear margin, resulting in an overall improved performance. Since VESC is an extension of Benoît et al. (2011), it is expected that both methods will perform similarly. However, one must note that the VESC is advanced in terms of computational cost, as we only learn one set of epitome and dictionary for the video denoising task, while on the other hand, for Benoît et al. (2011) approaches, it needs to learn the epitome and dictionary for each frame, which is much time consuming.

Tables 1 and 2 summarize the comparison between Benoît et al. (2011), Elad and Aharon (2006), Cheung et al. (2008) and VESC, using different noise levels, σ = {10, 20, 30, 40, 50} for the 'leaves' video, and $\sigma = \{15, 25\}$ for the 'tennis' video, respectively. In Table 1, it can be noticed that Cheung et al. (2008) has the lowest PSNR, while Benoît et al. (2011), Elad and Aharon (2006) and our proposed method have very competitive results. These quantitative findings are inline to our early qualitative results illustrated in Fig. 6. Similar results were achieved when using the 'tennis' video. As a summary, our approach outperform both methods (Cheung et al., 2008; Elad & Aharon, 2006) in all different noise levels, except when $\sigma = 20$ where the Elad and Aharon (2006) result is in par with our proposed method. These results have clearly shown that our method is significantly better than Cheung et al. (2008) and Elad and Aharon (2006) approaches, in both qualitatively and quantitatively. As expected, both the resultants from Benoît et al. (2011) and VESC are almost similar since the VESC is an extension from Benoît et al. (2011). However, in a much less computational cost, it makes the VESC a more favorable choice in video denoising task over the Benoît et al. (2011).

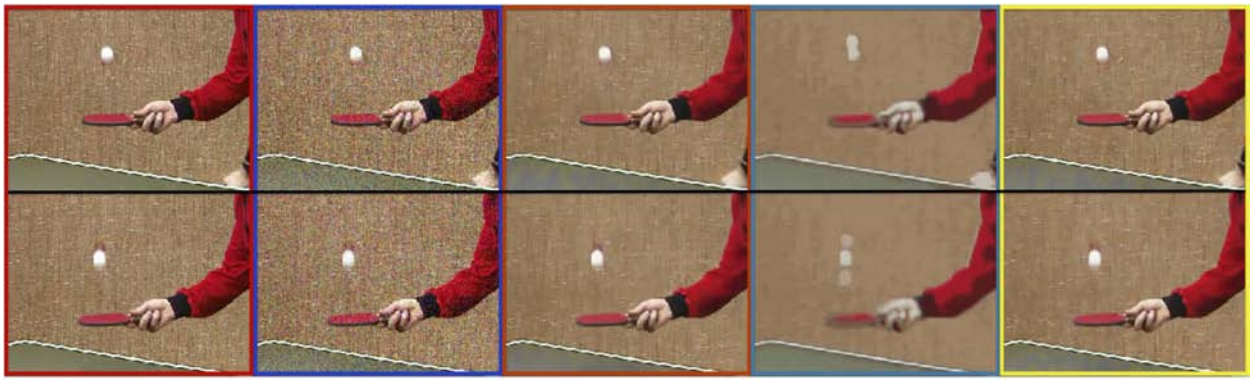
Comparison with other state-of-the-art methods: A comparison is performed using 6 video sequences, namely the 'bus', 'chair', 'football', 'renata', 'salesman', and 'tennis'. All these video frames are publicly available and we employed the color version, as similar to Dai et al. (2013). The comparison results are shown in Table 3. We specifically compare with MHMCF (Guo, Au, Ma, & Liang, 2007) and NLMC (Goossens, Luong, Aelterman, Pižurica, & Philips, 2010), as both of them are solutions in the spatial domain, similar to our proposed method. Besides, we also compare our methods with WRSTFC (Zlokolica, Pizurica, & Philips, 2005) which is a wavelet-based denoising method, LRGB and LRGB_{jme} which are the degraded version of CIFIC (Dai et al., 2013), LAYUV (Dai et al., 2010) which is an extension from the MHMCF, but in the transform domain, STGSM (Varghese & Wang, 2010) as the latest grayscale video denoiser, VBM3D (Dabov et al., 2007) as the latest grayscale image denoiser, and CIFIC that applied intercolor prediction in transform domain. From the results, we demonstrate that our proposed denoising system outperform all the spatial domain-based denoising methods (MHMCF and NLMC) when noise, $\sigma = 15$. Even in a higher noise setting ($\sigma = 25$), our proposed method still outperform the MHMCF by 1.78 db, and is comparable to NLMC with 0.12 db difference. In both noise settings, despite that solutions based on spatial domain are always lacked behind in compare to the transform-domain based solutions that are claimed to be more sophisticated, we still achieve comparable results as shown in the Table 3.

Specifically, a comparison with the current best implementation, the CIFIC (Dai et al., 2013), we only degrade by 1.65 db and 2.46 db in the $\sigma = \{15, 25\}$ settings, respectively. In overall, we achieve superior results in a less noisy setting, and comparable results on higher noisy settings, as compare to other state-of-the-art algorithms, quantitatively. Furthermore, we show a qualitative

(a) $N = 6$ (b) $N = 8$ (c) $N = 10$ (d) $N = 12$

Fig. 5. Visualization on different patch sizes, N on frame 1 of the 'leaves' video.

(a) 'leaves' video; $\sigma = 10$.



(b) 'tennis' video; $\sigma = 25$.

Fig. 6. Denoising results on 'leaves' and 'tennis' videos. From left to right: Original frame; noisy frame; KSVD; Epitome; VESC. First row indicates the frame 1 result, while the second row shows the frame 10 result.

Table 1
Result of our proposed method compared to video epitome (Cheung et al., 2008) and KSVD (Elad & Aharon, 2006) on 'tennis' video in PSNR (db).

Noise level, σ	15		25	
Noisy	24.78	–	20.41	–
Benoît et al. (2011)	30.43	+5.65	**27.39**	**+6.98**
Elad and Aharon (2006)	30.29	+5.51	27.09	+6.68
Cheung et al. (2008)	19.70	−5.08	20.01	−0.30
VESC	**30.44**	**+5.66**	**27.39**	**+6.98**

Bold values indicate best result.

comparison between state-of-the-art methods with our proposed method, for the 'tennis' sequence using $\sigma = 25$, in Figs. 8 and 9. Subjectively, we could visually notice there is a significant improvement of our method over the video epitome, and achieved similar performance to both the KSVD and NLMC in the Fig. 8. Please note that, despite the proposed method has a lower PSNR compare to the NLMC, we felt that the background patches (wall) in NLMC are over-smoothed compare to the original video, which lose the textures information. In our case, we manage to preserve the background texture better.

Then, in Fig. 9, we show our investigation of the denoising quality where a zoom into the bottom left region of frame 69 of the 'tennis' video is conducted. From the figure, it is noticed that the video epitome perform poorly in preserving the table tennis net texture. Meanwhile, our denoising system are still comparable to KSVD and NLMC, although losing to CIFIC in terms of preserving table tennis net texture. In conclusion, the visualization further validate our propose method advantage over video epitome and comparable to KSVD and NLMC.

4.1. Application to visual tracking

We also show using a simple tracking example that a good denoising algorithm is very important to enhance the performance of subsequent processes in the image processing domain. We employ the 'tennis' video and track the table tennis ball overtime. In this experiment, we prepare 3 different types of videos. The original video sequences serve as the groundtruth, a noisy video setting that has $\sigma = 25$, and a denoised video sequence using VESC. Adopted the tracking algorithm as to Lim, Chan, Monekosso, and Remagnino (2014a), the tracking results are as shown in Fig. 11. As expected, in a noisy video, the tracking algorithm fails due to the random noise (as shown in Fig. 11b), while in the denoising video using the VESC, the tracking results (in Fig. 11c) are comparable to the groundtruth (in Fig. 11a). From here, it shows the effectiveness of our denoising solution to handle the post-processing.

4.2. Discussions

Empirically, we have shown that the proposed VESC is capable to handle AWGN and significant to the state-of-the-art approaches in the spatial domain. Comparison to its variant – Benoît et al.

Table 2
Result of our proposed method compared to video epitome (Cheung et al., 2008) and KSVD (Elad & Aharon, 2006) on 'leaves' video in PSNR (db).

Noise level, σ	10		20		30		40		50	
Noisy	28.33	–	22.40	–	19.05	–	16.77	–	15.07	–
Benoît et al. (2011)	31.10	+2.77	**26.63**	**+2.23**	**23.46**	**+4.41**	**20.84**	**+4.07**	18.70	+3.63
Elad and Aharon (2006)	31.09	+2.76	**26.63**	**+2.23**	23.35	+4.30	20.78	+4.01	18.61	+3.54
Cheung et al. (2008)	16.70	−11.63	16.90	−5.50	16.98	−2.07	17.04	+0.27	16.94	+1.87
VESC	**31.12**	**+2.79**	**26.63**	**+2.23**	23.45	+4.40	20.83	+4.06	**18.71**	**+3.64**

Bold values indicate best result.

Table 3
Comparison to state-of-the-art denoising algorithms. Results on PSNR(db).

	MHMCF	NLMC	VESC	LRGB	$LRGB_{jme}$	LAYUV	WRSTFC	STGSM	VBM3D	$CIFIC_{2ref}$	$CIFIC_{3ref}$
σ = 15											
Bus	28.73	30.64	**31.36**	30.90	31.48	32.99	30.93	30.48	30.55	33.14	**33.47**
Chair	32.26	34.03	**34.55**	34.84	35.24	35.73	34.44	35.03	36.08	36.25	**36.40**
Football	27.77	29.77	**32.85**	29.58	29.98	31.02	N/A	29.88	30.57	31.33	31.59
Renata	29.37	31.04	**31.32**	31.95	32.31	32.90	31.19	32.69	32.76	33.14	**33.26**
Salesman	31.16	32.12	**32.29**	33.16	33.94	34.54	34.59	33.97	35.13	35.04	**35.18**
Tennis	29.80	**30.85**	30.44	31.14	31.71	32.49	31.38	31.29	32.38	32.58	**32.76**
Average	29.85	31.41	**32.13**	31.93	32.44	33.28	32.22	32.69	32.91	33.58	**33.78**
σ = 25											
Bus	25.67	**27.82**	27.77	27.73	28.56	29.98	28.25	27.73	27.81	30.15	**30.48**
Chair	29.19	**31.08**	30.69	32.17	32.82	33.18	31.80	32.70	**33.91**	33.60	33.71
Football	24.85	27.32	**29.84**	26.81	27.39	28.31	N/A	27.23	27.80	28.60	**28.82**
Renata	26.13	**27.94**	27.70	29.04	29.77	30.24	28.96	30.33	30.23	30.62	**30.74**
Salesman	27.64	**29.03**	27.43	30.00	31.09	31.49	32.04	30.66	**32.13**	31.93	32.10
Tennis	26.65	**28.36**	27.39	28.03	28.61	29.41	28.80	28.53	29.61	29.54	**29.74**
Average	26.69	**28.59**	28.47	28.96	29.71	30.44	29.97	29.53	30.25	30.74	**30.93**

Bold values indicate best result.

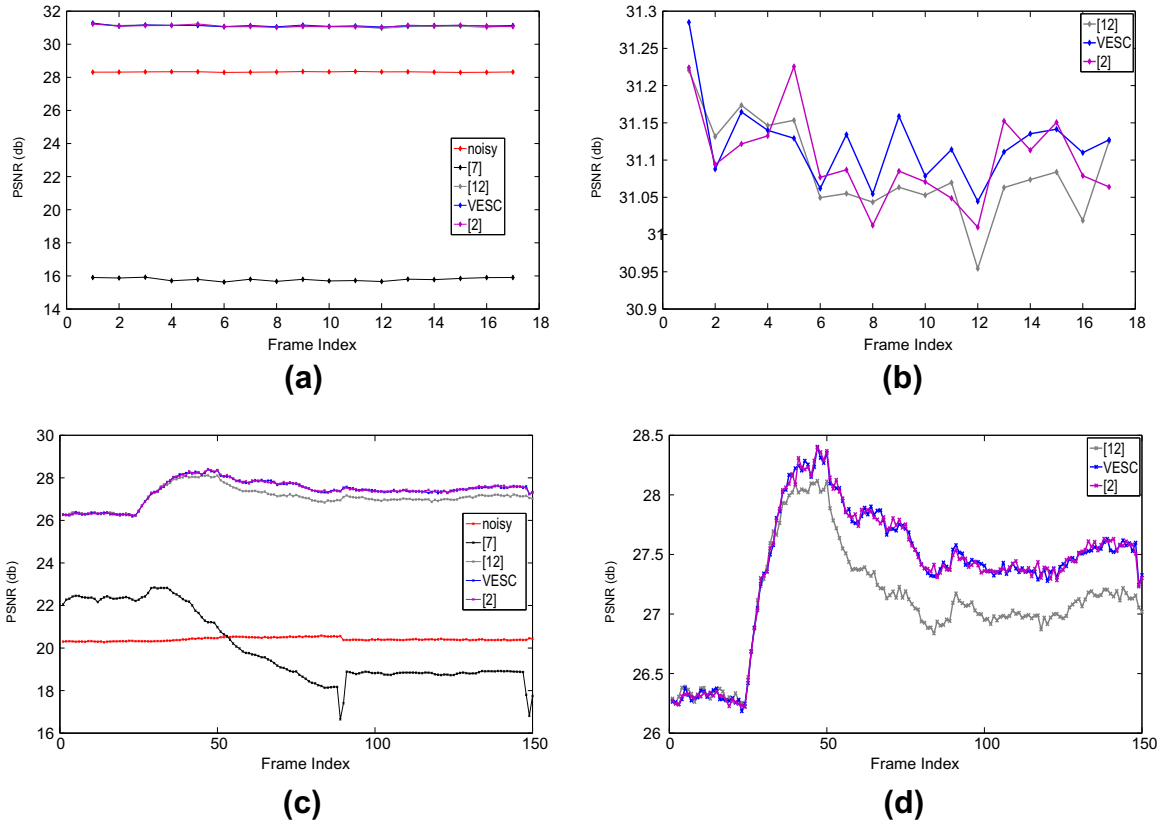


Fig. 7. (a) and (b): Frame-by-frame denoising results on 'leaves' video with $\sigma = 10$; (c) and (d): Frame-by-frame denoising results on 'tennis' video with $\sigma = 25$. The left column ((a) and (c)) shows the denoising results for video epitome (Cheung et al., 2008), KSVD (Elad & Aharon, 2006), image epitome with sparse coding (Benoît et al., 2011) and VESC in a noise-contaminated video; while the right column ((b) and (d)) is a zoom-in analysis to visualize the performance gap for KSVD (Elad & Aharon, 2006), image epitome with sparse coding (Benoît et al., 2011), and VESC, respectively.

(d) NLMC (e) CIFIC (f) VESC

Fig. 8. Visualization of denoising results using different methods on 'tennis' sequence frame 69, and noise level, $\sigma = 25$.

(a) Original (b) Video epitome (c) KSVD (d) NLMC (e) CIFIC (f) VESC

Fig. 9. Visualization of denoising results using different methods on 'tennis' sequence frame 69, which is zoomed into bottom left corner, and noise level, $\sigma = 25$.

(a) Original (b) Noisy (c) VESC denoise

Fig. 10. Example of the denoising results with Poisson noise at frame 38.

(a) Original (b) Noisy (c) Denoise

Fig. 11. Visual tracking in tennis sequence for frame 38. (a) is the original frame (as groundtruth); (b) is contaminated by AWGN with $\sigma = 25$; (c) is the tracking result after denoised by VESC. It can be noticed that the tracking algorithm fails to track the ball in the noisy image.

(2011), we also shown that the proposed VESC is advanced in terms of computational cost as the Benoît et al. (2011) require one epitome and one KSVD dictionary for each frame in a video clip for video denoising, while the VESC unified one epitome and one KSVD dictionary for the entire video clip. However, one limitation in the VESC is that it does not cope well with non-Gaussian noise. For example. we apply Poisson noise into the 'leaves' video, and show the denoising results in Fig. 10. As to our expectation, the

VESC does not able to cope well in the Poisson noise – a type of non-Gaussian noise. We anticipate that similar results will be achieved using other non-Gaussian noise such as impulse noise, multiplicative noise etcs. However, one must note that denoising in non-Gaussian noise is a research by its own domain. For instance, hand-crafted solution is essential such as Gaussianizing the Poisson measurements as indicated by Luisier, Blu, and Unser (2011). In this case, the VESC would require a new dictionary learning method as to Ma, Moisan, Yu, and Zeng (2013) to perform well in the non-Gaussian noise. However, the aim of the paper is to show how Benoît et al. (2011) – a spatial domain, image denoising approach could be extended to video denoising approach, in a unified framework. That is, how to learn an epitome and a KSVD dictionary for entire video clip to perform denoising, and hence denoising in the non-Gaussian noise will be considered as our future work. Also, worth to mention that most of the solutions in the non-Gaussian noise are in the transform domain, while the VESC is in the spatial domain therefore the extension is not direct.

5. Conclusion

This paper presented a compact representation for a video denoising system, using the epitomic-based dictionary learning structure. In specific, we use the video epitome to build the compact representation on noisy video, and learn the KSVD dictionary from the video epitome to denoise the noisy video. The proposed method is designed for the AGWN noise, and is a spatial domain solution. In overall, the proposed method shows convincing results over the baseline methods, as well as the state-of-the-art methods that within the spatial domain. Besides, the proposed method manage to reduce the computational cost on video denoising compare to the most related method (Benoît et al., 2011), that require to denoise based on frame-by-frame basis. However, the proposed method has a limitation to work on the non-Gaussian noise. As an example in the Poisson noise, our current solution will have significant information lose on the background information.

One of the future works will consist of improving the efficiency of the algorithms with an introductory of multiple epitomes extension. Besides that, we will test our algorithm in different test sequences with different complexity, as well as higher range of σ. On top of that, it will be interesting to extend the current framework to non-Gaussian noise e.g. impulse noise, Poisson noise and multiplicative noise. Finally, as shown in the experiment where a good denoising solution will enhance the performance of subsequent processes; therefore, we are also interested to investigate how denoising methods can enhance subsequent processes such as compression, segmentation, object recognition and detection.

Acknowledgment

This research is supported by the High Impact MoE Grant UM.C/625/1/HIR/MoE/FCSIT/08, H-22001-00-B00008 from the Ministry of Education Malaysia.

References

Aharon, M., & Elad, M. (2008). Sparse and redundant modeling of image content using an image-signature-dictionary. *SIAM Journal on Imaging Sciences, 1*, 228–247.

Benoît, L., Mairal, J., Bach, F., & Ponce, J. (2011). Sparse image representation with epitomes. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 2913–2920).

Blu, T., & Luisier, F. (2007). The sure-let approach to image denoising. *IEEE Transactions on Image Processing, 16*, 2778–2786.

Boulanger, J., Kervrann, C., Bouthemy, P., Elbau, P., Sibarita, J.-B., & Salamero, J. (2010). Patch-based nonlocal functional for denoising fluorescence microscopy image sequences. *IEEE Transactions on Medical Imaging, 29*, 442–454.

Buades, A., Coll, B., & Morel, J.-M. (2005). A review of image denoising algorithms, with a new one. *Multiscale Modeling & Simulation, 4*, 490–530.

Chan, C. S., & Liu, H. (2009). Fuzzy qualitative human motion analysis. *IEEE Transactions on Fuzzy Systems, 17*, 851–862.

Cheung, V., Frey, B. J., & Jojic, N. (2008). Video epitomes. *International Journal of Computer Vision, 76*, 141–152.

Dabov, K., Foi, A., & Egiazarian, K. (2007). Video denoising by sparse 3d transform-domain collaborative filtering. In *European signal processing conference (eusipco)* (p. 7). Vol. 1.

Dai, J., Au, O. C., Pang, C., & Zou, F. (2013). Color video denoising based on combined interframe and intercolor prediction. *IEEE Transactions on Circuits and Systems for Video Technology, 23*, 128–141.

Dai, J., Au, O. C., Yang, W., Pang, C., Zou, F., & Wen, X. (2010). Color video denoising based on adaptive color space conversion. In *IEEE international symposium on circuits and systems (ISCAS)* (pp. 2992–2995).

Eksioglu, E. M. (2014). Online dictionary learning algorithm with periodic updates and its application to image denoising. *Expert Systems with Applications, 41*, 3682–3690.

Elad, M., & Aharon, M. (2006). Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing, 15*, 3736–3745.

Goncalves, W. N., & Bruno, O. M. (2013). Dynamic texture analysis and segmentation using deterministic partially self-avoiding walks. *Expert Systems with Applications, 40*, 4283–4300.

Goossens, B., Luong, H., Aelterman, J., Pižurica, A., & Philips, W. (2010). A gpu-accelerated real-time nl means algorithm for denoising color video sequences. In *Advanced concepts for intelligent vision systems* (pp. 46–57).

Guo, L., Au, O. C., Ma, M., & Liang, Z. (2007). Temporal video denoising based on multihypothesis motion compensation. *IEEE Transactions on Circuits and Systems for Video Technology, 17*, 1423–1429.

Jojic, N., Frey, B.J., & Kannan, A. (2003). Epitomic analysis of appearance and shape. In *International conference on computer vision (ICCV)* (pp. 34–41).

Karasulu, B., & Korukoglu, S. (2012). Moving object detection and tracking by using annealed background subtraction method in videos: Performance optimization. *Expert Systems With Applications, 39*, 33–43.

Kuang, Y., Zhang, L., & Yi, Z. (2014). An adaptive rank-sparsity k-svd algorithm for image sequence denoising. *Pattern Recognition Letters, 45*, 46–54.

Lazebnik, S., Schmid, C., & Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 2169–2178). Vol. 2.

Lim, M. K., Chan, C. S., Monekosso, D., & Remagnino, P. (2014a). Refined particle swarm intelligence method for abrupt motion tracking. *Information Sciences, 283*, 267–287.

Lim, M. K., Tang, S., & Chan, C. S. (2014b). isurveillance: Intelligent framework for multiple events detection in surveillance videos. *Expert Systems with Applications, 41*, 4704–4715.

Luisier, F., Blu, T., & Unser, M. (2011). Image denoising in mixed Poisson–Gaussian noise. *IEEE Transactions on Image Processing, 20*, 696–708.

Mairal, J., Elad, M., & Sapiro, G. (2008). Sparse representation for color image restoration. *IEEE Transactions on Image Processing, 17*, 53–69.

Ma, L., Moisan, L., Yu, J., & Zeng, T. (2013). A dictionary learning approach for poisson image deblurring. *IEEE Transactions on Medical Imaging, 32*, 1277–1289.

Milanfar, P. (2013). A tour of modern image filtering: New insights and methods, both practical and theoretical. *IEEE Signal Processing Magazine, 30*, 106–128.

Olshausen, B. A., & Field, D. J. (1997). Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision Research, 37*, 3311–3325.

Papadopoulos, D. P., Kalogeiton, V. S., Chatzichristofis, S. A., & Papamarkos, N. (2013). Automatic summarization and annotation of videos with lack of metadata information. *Expert Systems with Applications, 40*, 5765–5778.

Peyré, G. (2009). Sparse modeling of textures. *Journal of Mathematical Imaging and Vision, 34*, 17–31.

Protter, M., & Elad, M. (2009). Image sequence denoising via sparse and redundant representations. *IEEE Transactions on Image Processing, 18*, 27–35.

Rao, Y., & Chen, L. (2012). A survey of video enhancement techniques. *Journal of Information Hiding and Multimedia Signal Processing, 3*, 71–99.

Rubinstein, R., Zibulevsky, M., & Elad, M. (2010). Double sparsity: Learning sparse dictionaries for sparse signal approximation. *IEEE Transactions on Signal Processing, 58*, 1553–1564.

Varghese, G., & Wang, Z. (2010). Video denoising based on a spatiotemporal gaussian scale mixture model. *IEEE Transactions on Circuits and Systems for Video Technology, 20*, 1032–1040.

Wang, X.-Y., Yang, H.-Y., & Fu, Z.-K. (2010). A new wavelet-based image denoising using undecimated discrete wavelet transform and least squares support vector machine. *Expert Systems with Applications, 37*, 7040–7049.

Wu, Z., Cao, J., Tao, H., & Zhuang, Y. (2013). A novel noise filter based on interesting pattern mining for bag-of-features images. *Expert Systems with Applications, 40*, 7555–7561.

Yang, R., & Ren, M. (2011). Wavelet denoising using principal component analysis. *Expert Systems with Applications, 38*, 1073–1076.

Zlokolica, V., Pizurica, A., & Philips, W. (2005). Wavelet based motion compensated filtering of color video sequences. In *Optics & photonics 2005* (pp. 59141P–59141P).