Available online at www.sciencedirect.com

SCIENCE DIRECT®

Image and Vision Computing 24 (2006) 143–152

www.elsevier.com/locate/imavis

Video dejittering by bake and shake

Sung Ha Kang [a,*], Jianhong Shen [b]

[a] *Department of Mathematics, University of Kentucky, 715 Patterson Office Tower, Lexington, KY 40506, USA*
[b] *School of Mathematics, University of Minnesota, 206 Church St. SE, Minneapolis, MN 55455, USA*

Received 24 September 2004; accepted 20 September 2005

Abstract

Video jittering occurs when the horizontal lines of video image frames are randomly displaced due to the corruption of synchronization signals or electromagnetic interference during video transmission. Inspired by the recent Bayesian/variational dejittering model of Shen (*SIAM J. Appl. Math., vol*. 64, pp. 1691-1708, 2004), in the current paper we propose a novel dejittering approach nicknamed 'bake-and-shake.' The *bake* step is to apply Perona–Malik type nonlinear diffusions to 'melt away' or heat up the jittered video frames, based upon which the *shake* step is able to optimally estimate the individual line jitters and renormalize the jittered images. Numerical implementation of the bake-and-shake algorithm as well as several computational results are presented.

Keywords: Line jitters; Dejittering; Perona–Malik; Bake; Edge preservation; Total variation; PDE based method; Newton–Raphson; Shake; Texture

1. Introduction

Video jittering occurs when the synchronization signals on analog video tapes axe corrupted or when environmental electromagnetic interference randomly delays video signals. It can often result in random displacements (i.e. jitters) of horizontal lines in video image frames, and make video images either unpleasant to watch or difficult to perceive important visual features. Fig. 1 shows a typical jittered video frame.

Traditionally, jittering in analog video tapes has been corrected by retrieving the signal information on tapes that has nothing to do with image contents. Such dejittering approaches are thus *nonintrinsic* since they fail to work properly if only provided with the jittered video images alone.

In the remarkable works [1,2], Kokaram et al. first explored dejittering methods that only rely on the video images without demanding any extra irrelevant information. The authors proposed to register shuffled lines using multiresolution schemes and autoregressive image models for *intrinsic* video dejittering. Following the same line of *intrinsic dejittering*, in Ref. [3] Shen developed a single variational dejittering model based on the Bayesian rationale and the celebrated BV image model of Rudin, Osher, and Fatemi [4].

Inspired by the independent meaning of the Euler–Lagrange Equations of the variational model of Shen [3], in the present work, we propose a flexible two-step method for image dejittering, which we have nicknamed 'bake and shake.' Likewise in all the aforementioned works [1–3], this new approach also shares two characteristics. (a) First, it is also information-theoretically intrinsic and does not depend upon any extra information (e.g. storage media or starting signals) other than the input jittered video frames. (b) It is also *intra-frame* intrinsic and does not require the knowledge of correlations among different video frames. The latter attribute reduces the video dejittering problem to a image estimation and restoration problem.

As the name explicitly suggests, the proposed bake-and-shake dejittering approach consists of two main ingredients: bake and shake.

The bake step is to employ Perona–Malik type nonlinear diffusions [5], that can 'heat up' the jittered video images and 'melt away' the jigsaw patterns caused by horizontal random line jitters. By doing so, on one hand, the jittered images are inevitably mollified after losing many details, but on the other hand, which is even more crucial, vertical correlations among randomly shuffled horizontal lines become manifest in the baked images. It is the valuable positional information unveiled by this vertical correlation that will greatly facilitate the next step of shaking. We shall explain in details why

* Corresponding author. Tel.: +1 859 257 6815; fax: +1 859 257 4078.
E-mail addresses: skang@ms.uky.edu (S.H. Kang), jhshen@math.umn.edu (J. Shen).

0262-8856/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2005.09.022

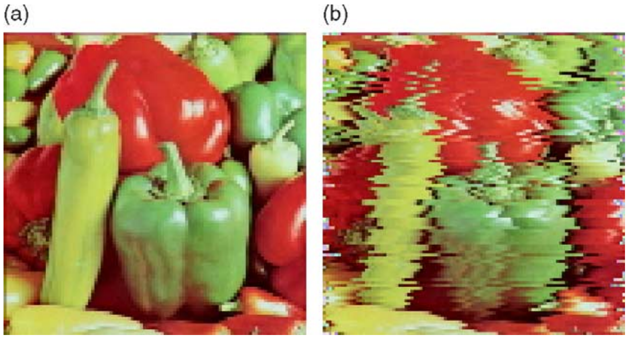

Fig. 1. (a) The ideal image. (b) The jittered image.

Perona–Malik's anisotropic diffusions can lose less information during the baking step (compared with linear diffusions) and lead to more robust dejittering performance.

The shake step follows immediately the bake step (see Fig. 2), and is the major tool for estimating the random jitters. The vertical correlation revealed by the bake step helps the shake step to optimally identify the random jitters along all the individual horizontal lines, and to reshuffle (or *shake*) the jittered image. Due to ubiquitous intensity noises, the shake step in our work is formulated as a least-square estimation problem, and computed based on the Newton–Raphson iterative algorithm for nonlinear optimization [6].

Furthermore, this two-step machinery towards dejittering can be cascaded when necessary, even though our computational results show that one round of bake and shake often already suffices for many applications.

Finally, the remarkable flexibility of our proposed bake-and-shake approach also manifests in its versatile handling of color images (or multichannel images more generally). The bake step can be easily implemented using numerous existent nonlinear diffusion models for color or multichannel images (see, e.g. [7–12])

The organization is as follows. In the five subsections of Section 2, we explain the details of the bake-and-shake model and algorithm. We start with the general statistical formulation of the jittering problem. After revealing the major functions of the bake and shake mechanisms, we explain separately the mathematical formulations of each bake and shake steps. In particular, we thoroughly explain the advantages of Perona–Malik's nonlinear diffusions for the bake step, comparing with conventional linear heat diffusions. Section 3 consists of two subsections on the computation of the bake-and-shake model and algorithm. The first subsection details all the implementation schemes for both the bake and shake steps, while the second subsection contains several computational examples to illustrate the performance of our proposed bake-and-shake approach. Conclusion and future works are discussed in Section 4.

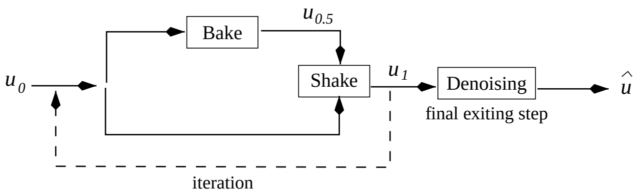

Fig. 2. Flow chart of the bake-and-shake algorithm.

2. The proposed dejittering model: bake and shake

2.1. The jittering problem

We first briefly discuss the stochastic model of the jittering process. Let $\Omega=(a,b)\times(0,H)\subset\mathbb{R}^2$ be a Cartesian square domain modeling the continuum limit of most contemporary display devices such as the monitors of TV's, computers, or cameras. A typical pixel in Ω will be denoted by (x, y). For technical convenience, it shall also be assumed that $a=-\infty$ and $b=+\infty$. (For finite a and b, boundary conditions can be properly formulated as in Ref. [3].)

Let $u_{\text{ideal}}(x, y)$ be an ideal image without jittering on Ω. For each $y\in(0, H)$, there exists some random jitter $s(y)\in\mathbb{R}$, which is often assumed to be Gaussian with mean zero. In addition, $s(y)$'s are independent for different y's, and can further be assumed homogeneous. In combination, these lead to the Gaussian white noise model for random horizontal jitters $s(y)$ [3]. More general jittering models can be developed based upon Markov chains or other deterministic regularities [3].

In addition, as common in many applications, intensity noises are also ubiquitous due to aged film grains or electromagnetic fluctuations in the environment. Assume that the intensity noise $n(x, y)$ involved is additive and has zero mean. One can also further assume the intensity noise is Gaussian white, which is, however, not essential for the present work.

Thus, the noisy image observation $u_0(x, y)$ under random horizontal jittering can be modeled by

$$u_0(x,y) = u_{\text{ideal}}(x-s(y),y) + n(x,y), \quad x,y\in\Omega.$$

By the stochastic nature of both $s(y)$ and $n(x, y)$, strictly speaking, $u_0(x, y)$ becomes a random field. In reality, however, $u_0(x, y)$ stands for a typical (in the information theoretical sense [13]) sample of the associated random field. The goal of practical dejittering is to restore the ideal image $u_{\text{ideal}}(x, y)$ based on a typical single observation $u_0(x, y)$.

2.2. Overview of the bake-and-shake algorithm

In the current paper, we propose a two-step model for the dejittering problem, which has been nicknamed the *bake-and-shake* process.

We shall explain the two steps in great details in the coming two subsections. Here, we first outline the main mechanisms underlying this novel dejittering process. Fig. 2 shows the flow chart of this algorithm. The bake-and-shake process can be iterated when necessary.

The *bake* step applies a diffusion process to the noisy and jittered image u_0. Its action bears twofold missions. First, as well known in the image processing literature, diffusion is a filtering process that can suppress the effect of intensity noises *n*. Second, for the jittered image u_0, diffusion also 'melt away' the jigsaw object boundaries introduced by random horizontal shuffling. It works well mainly because the diffusion is a low-pass filtering processing while jittered boundaries are high-frequency features. Thus, the net effect of the bake step is to produce a mollified (or *baked*) intermediate image u (denoted by $u_{0.5}$ in Fig. 2) that substantially suppresses the effects of both the intensity noise $n(x, y)$ and the horizontal jittering $s(y)$. However, object boundaries in the baked image become inevitably blurry, and typically still contain the remanent low-frequency components from the jittering process. This necessitates the shake step.

The *shake* step estimates the jitter variable $s(y)$ based upon both the original image u_0 and the output image u from the bake step. It treats u as a good approximation to the ideal image and employs a least-square criterion to match $f_0(x) = u_0(x, y)$ with $f(x+s) = u(x+s(y), y)$ for each horizontal line $y \in (0, H)$. From the estimated jitters $s(y)$, the output of the shake step is $u_1(x, y) = u_0(x\text{-}s(y), y)$, which shakes up the jitters and makes the image almost jitter free. Notice that u_1 still contains the intensity noise, which can be cleaned up using any classical denoising schemes (see Fig. 2) [4,5,14,15].

2.3. The bake step

The importance of the bake step is to introduce vertical communication among the horizontal lines. This is illustrated in Fig. 3. Panel (a) shows a synthetic ideal image u_{ideal}, (b) its noisy and jittered version u_0, and (c) the baked image u. The second row shows the profiles of their 27th rows separately. The noisy profile of image (b) is more than 5 pixels (which is quite large in the literature [16]) off to the right, whereas the profile of the baked is very close to the ideal image in terms of horizontal positioning.

The core mechanism of the bake step is the diffusion process, which includes the ordinary heat diffusion as a familiar example. In this physics sense, the bake step acts upon the jittered noisy image u_0 in very much the same way as heat melts away ice tips. The net effect of the diffusion-based bake step is that random horizontal jitters are mollified and the average smooth boundaries resurface in the baked image, as manifest in the example of Fig. 3.

Mathematically, the jittering process introduces high-frequency components into the image, whereas diffusion is a low-pass filtering process which gradually suppresses high-frequency irregularities. Diffusion has thus been widely applied in image analysis and processing (see, e.g. Witkin [17] and the mono-graph by Weickert [18]). Normally diffusion is only employed to remove intensity noises, here it additionally (and more importantly) helps regularize the shuffled boundaries.

Among many alternatives of diffusion processes, in the present work, we have employed the celebrated anisotropic diffusion model proposed by Perona and Malik [5]:

$$\frac{\partial u}{\partial t} = \nabla \cdot [g(|\nabla u|)\nabla u], \quad u|_{t=0} = u_0. \tag{1}$$

Here, the nonlinear diffusivity $g(p)$ of $p = |\nabla u|$ should *at least* satisfy

$$g(0) > 0, \quad g(p) \text{ decreases as } p \to \infty, \quad \text{and } g(+\infty) = 0.$$

Common choices include, for example, the Cauchy function

$$g(p) = \frac{1}{1 + p^2/a^2}, \quad \text{for some } a > 0,$$

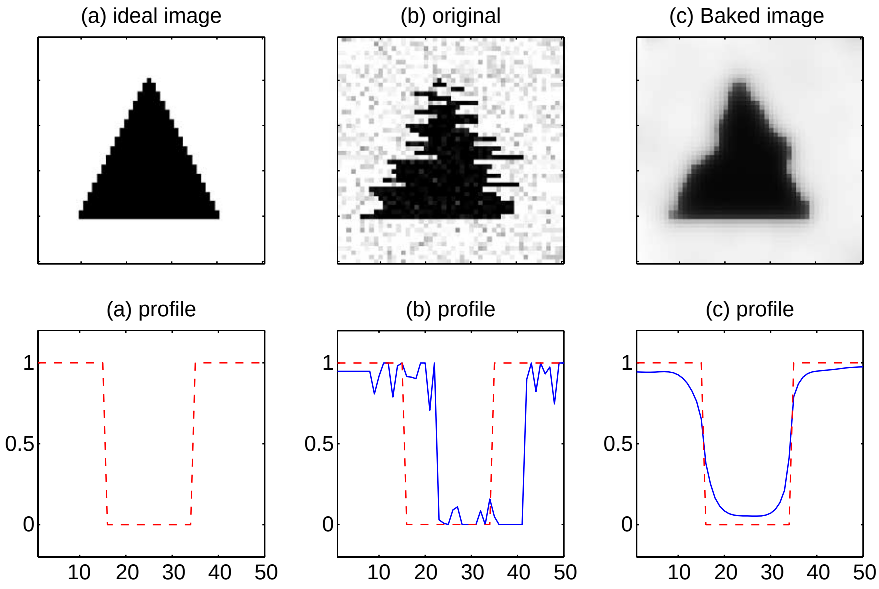

Fig. 3. (a) The ideal image, (b) the given noisy and jittered image u_0, (c) the baked image u. Plotted in the second row are the corresponding profiles of the 27th row. The (red) dashed line is the profile of the ideal image, while the (blue) solid lines are those associated with (b) and (c). It is evident that the profile of the baked image is horizontally much closer to the ideal one, though smoothened to some extent. Baking has thus promoted vertical communication among the horizontal lines, so that each one can approximately resume to its original position when assisted by the others in the vicinity.

the Guassian function

$$g(p)=\exp\left(-\frac{p^2}{a^2}\right),\quad \text{for some } a>0,$$

the minimal-surface diffusivity

$$g(p)=\frac{1}{\sqrt{1+p^2}},\quad \text{or more generally,}\quad g(p)=\frac{1}{\sqrt{a^2+p^2}},$$

and the diffusivity inspired by the total variation Radon measure [4] and mean curvature motions $g(p)=1/p$.

Due to the numerical smoothing effect of finite difference schemes, Eq. (1) is usually well posed computationally. Nevertheless, the noisy initial data u_0 stir up theoretical difficulty in justifying the initial computation of the gradient ∇u. This issue can be resolved by properly regularizing the diffusivity from

$$g(|\nabla u|)\text{ to } g(|\nabla u_\sigma|),$$

where $u_\sigma = G_\sigma * u$ stands for the mollified version of u using a thin and tall Gaussian kernel with small variance σ^2. Interested readers are referred to the important work of Catté, Lions, Morel, and Coll [19].

Unlike conventional *linear* diffusions, Perona–Malik diffusions cleverly introduce adaptivity into the diffusivity coefficients: faster diffusion on homogeneous regions where images are smoother, and slower near edges where images gradients become large. This edge-preservation property makes Perona–Malik diffusions ideal for the dejittering application, since the edge information will play a crucial role in robust shaking. We shall further elaborate on this point in a quantified and precise manner in Section 2.5.

2.4. The shake step

Let u denote the output from the bake step, which is a blurry approximation to the ideal image. As illustrated in Fig. 3, u and the ideal image share the similar positional information, and consequently, u can be employed to estimate the jitters, which is the task of the shake step.

For any fixed $y\in(0,H)$, define $f_0(x)=u_0(x,y)$ and $f(x)=u(x,y)$, which are horizontal slices as illustrated in Fig. 4. We are interested in finding the optimal jitter s that minimizes the following error function

$$e(s)=\int_{\mathbb{R}}(f_0(x)-f(x+s))^2\mathrm{d}x,$$

where as in Section 2.1, the image domain is assumed to be $\Omega=\mathbb{R}\times(0,H)$ for simplicity (see Shen [3] for details in coping with the boundaries of bounded domains). The optimal line jittering s must satisfy $e'(s)=0$, where

$$e'(s)=-\int_{\mathbb{R}}(f_0(x)-f(x+s))\cdot f'(x+s)\,\mathrm{d}x.$$

which is a 1D nonlinear optimization problem.

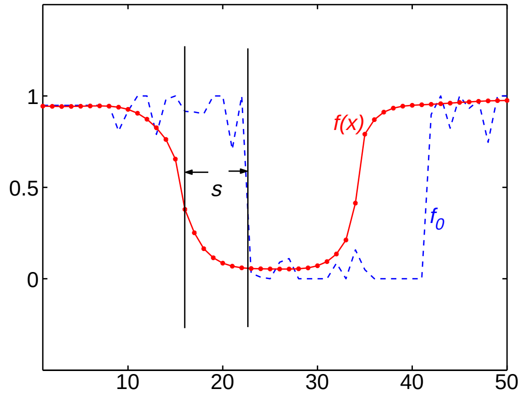

Fig. 4. Following the bake step in Fig. 3, for any fixed $y\in(0,H)$, $f_0(x)$ is the horizontal profile of the original noisy and jittered image $u_0(x,y)$, while $f(x)$ is that of the baked (diffused) image $u(x,y)$. The displacement s approximates the ideal unknown jitter t.

We then employ the Newton–Raphson scheme to iteratively find the optimal jitter estimation:

$$s_{n+1}=s_n-\frac{e'(s_n)}{e''(s_n)},\quad \text{often with } s_0=0.$$

Let $\langle f,g\rangle$ denote the inner product in the Hilbert space $L^2(\mathbb{R})$. Then the first derivative becomes $e'(s)=\langle f_0(x)-f(x+s),f'(x+s)\rangle$. Assuming the profiles die out in the infinities (or the Neumann boundary condition for finite domains), the second derivative becomes

$$\begin{aligned}e''(s)&=\left\langle f'(x+s),f'(x+s)\right\rangle-\left\langle f_0(x)-f(x+s),f''(x+s)\right\rangle\\&=-\left\langle f_0(x),f''(x+s)\right\rangle,\end{aligned}$$

where the last equality results from integration by parts:

$$\left\langle f(x+s),f''(x+s)\right\rangle=-\left\langle f'(x+s),f'(x+s)\right\rangle.$$

Assume that the profile of the original image $f_0(x)$ has been generated by the true jitter t, i.e. $f_0(x)=f_{\text{ideal}}(x+t)+n(x)$, where f_{ideal} denotes the ideal profile. Suppose the baked profile $f(x)$ is close to $f_{\text{ideal}}(x)$, then

$$f_0(x)=f_{\text{ideal}}(x+t)+n(x)\approx f(x+t)+n(x).\tag{2}$$

Since $n(x)$ is independent of the image signal, one has $\langle n(x),f''(x+s)\rangle=0$. Thus, the second derivative $e''(s)$ can be expressed as

$$e''(s)=-\left\langle f_0(x),f''(x+s)\right\rangle\approx-\left\langle f(x+t),f''(x+s)\right\rangle,$$

which, based on integration by parts, yields

$$e''(s)\approx\left\langle f'(x+t),f'(x+s)\right\rangle.$$

Applying Taylor expansion to $f'(x+t)$,

$$f'(x+t)=f'(x+s)+f''(x+s)(t-s)+O(|t-s|^2).$$

(Notice that the smoothness requirement for Taylor expansion is indeed satisfied since f is the baked (or mollified) version.) Furthermore, since

$$\langle f''(x+s), f'(x+s)\rangle = \frac{(f'(x+s))^2}{2}\Big|_{-\infty}^{\infty} = 0$$

(either from the vanishing conditions at ± ∞ or the Neumann boundary condition for finite domains), one eventually has

$$e''(s) \approx \langle f'(x+s), f'(x+s)\rangle + O(|t-s|^2),$$

where t denotes the unknown true jitter. Due to the shift-invariance of Lebesgue measures, one can replace the original denominator $e''(s)$ in the Newton–Raphson algorithm by its approximate version

$$e''(s) \approx \|f'(x+s)\|^2 = \|f'(x)\|^2,$$

which is in fact independent of s_n, and lessens the computational costs.

In conclusion, the Newton–Raphson scheme can be well approximated by

$$\begin{aligned} s_{n+1} &= s_n - \frac{e'(s_n)}{e''(s_n)} \\ &= s_n + \frac{\langle f_0(x) - f(x+s_n), f'(x+s_n)\rangle}{\|f'(x)\|^2}, \end{aligned} \qquad (3)$$

which is more robust since the denominator is guaranteed to be nonzero, (unless the image profile is trivial and featureless, i.e. $f(x) \equiv$ some constant).

The new approximate scheme Eq. (3) permits an independent interpretation. Suppose the true jitter $t \ll 1$ is small, then a single iteration almost suffices for accomplishing the shake step. Starting with the initial guess $s_0 = 0$ (since $t \ll 1$), one has

$$\begin{aligned} s_1 &= s_0 + \frac{\langle f_0(x) - f(x+s_0), f'(x+s_0)\rangle}{\|f'(x)\|^2} \\ &= \frac{\langle f_0(x) - f(x), f'(x)\rangle}{\|f'(x)\|^2}. \end{aligned} \qquad (4)$$

By Eq. (2), Taylor expansion leads to

$$f_0(x) \simeq f(x+t) + n(x) = f(x) + n(x) + tf'(x) + O(t^2).$$

Therefore,

$$s_1 = t + O(t^2).$$

As previously explained, the error can be further refined to $O(t^3)$ if $f(x)$ well approximates the ideal profile (which is valid after several iterations of the bake-and-shake algorithm).

2.5. Why Perona–Malik baking for robust shaking

With a help from the shake step in Section 2.4, we now explain the intrinsic advantages of Perona–Malik diffusions over linear isotropic ones for the bake step.

First, as highlighted before, the main goal of the bake step is to promote vertical communication among randomly shuffled horizontal slices, so that their average positioning information is extracted as faithfully as possible. (This of course comes at the necessary cost of blurring and mollifying the given image.) Fig. 5 vividly demonstrates the advantage of Perona–Malik baking over linear-diffusion based baking.

In addition to more efficiently recovering the original positioning information, Perona–Malik baking also makes the Newton–Raphson shaking more stable and robust, as manifest in the denominator $\|f'(x)\|^2$ in Eq. (3). With an assistance from Fig. 6, this can be independently quantified as follows. The top panel shows a *constant* (1D) image signal $f(x)$ and its jittered version $f_0(x) = f(x+t)$ for some fixed positive jitter t. The featureless nature, especially lack of the edge information, of the signal leaves no jittering clue for the dejittering. In fact any jitter estimation s works for this example. Notice that this is an extreme featureless image with $\|f'\| = 0$. The second signal $f(x)$ in the middle panel is nonconstant, and the difference between $f_0(x)$ and $f(x)$ gives certain information about the jitter. Consider the error function $e(s) = \|f_0(x) - f(x+s)\|^2$, where

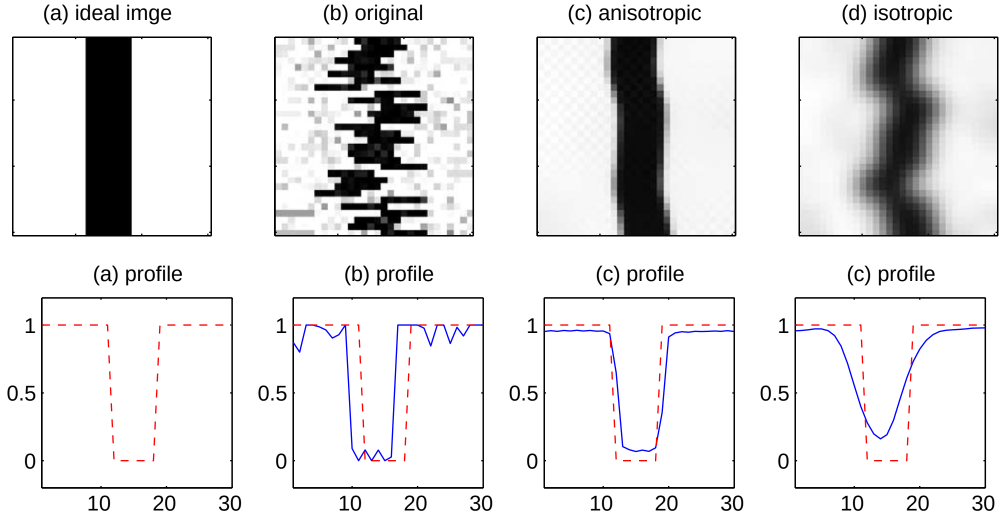

Fig. 5. (a) The ideal image, (b) the noisy and jittered version u_0, (c) the baked image u by Perona–Malik anisotropic diffusion, (d) the baked image u by a linear diffusion process. The second row shows their corresponding horizontal profiles for a typical y. The (red) dashed line denotes the ideal image profile, while the (blue) solid lines are the individual profiles. Notice that Perona–Malik baking recovers the original positioning information much more faithfully than linear diffusions.

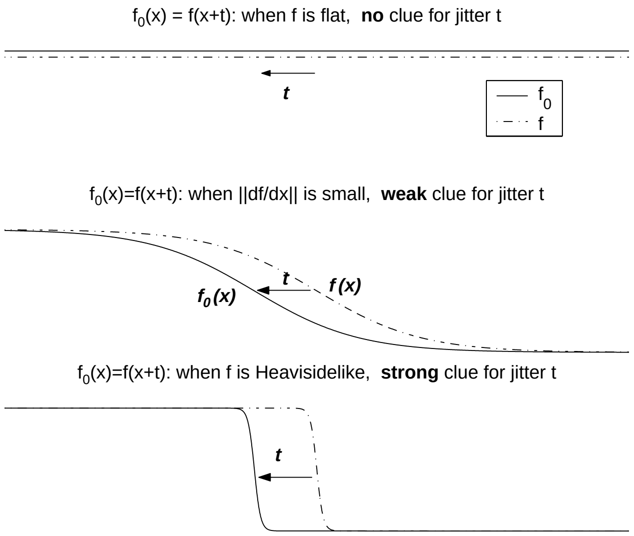
f0(x) = f(x+t): when f is flat, no clue for jitter t
t
f0
f
f0(x)=f(x+t): when ||df/dx|| is small, weak clue for jitter t
t
f(x)
f0(x)
f0(x)=f(x+t): when f is Heavisidelike, strong clue for jitter t
t

Fig. 6. Role of sharp edges for robust dejittering—illustrated via a noiseless 1D horizontal slice: $f_0(x)=f(x+t)$ with jitter $s=t$: larger slopes or larger $\|f'(x)\|$ norms lead to more robust jitter estimation.

$f_0(x)=f(x+t)$ with t unknown and to be estimated. For any estimator s in the vicinity of the true jitter t, Taylor expansion yields

$$e(s) = \|f(x+t)-f(x+s)\|^2 \approx \|f'(x+t)(t-s)\|^2$$
$$= \|f'(x)\|^2|t-s|^2.$$

With $\|f'(x)\|$ weak enough, this formula implies the difficulty in reading the jittering information from the error function $e(s)$ since it is insensitive to the positional error $|t\text{-}s|$. Finally, the bottom panel shows a signal $f(x)$ close to the ideal Heaviside step edge $H(x)$:

$$H(x) = \begin{cases} 1, & x<0 \\ 0, & \text{otherwise} \end{cases}.$$

The error function is given by

$$e(s) = \int_{-\infty}^{\infty} |H(x+t)-H(x+s)|^2 \, \mathrm{d}t = |t-s|.$$

In this ideal case, the jitter information can be directly readable from the error function!

To conclude, it is desirable for the bake step to be able to preserve sharp edges, and large gradient information makes the shake step more robust. This makes Perona–Malik diffusions more appealing than linear ones (see Fig. 5).

3. Numerics and computational results

In this section, we first present the numerical details for implementing the bake-and-shake algorithm, and then demonstrate the performance of our proposed model and algorithm through several generic examples.

3.1. Numerical details for implementing bake and shake

For the bake Eq. (1), we take $g(|\nabla u|)=g(p)=1/\sqrt{p^2+a^2}$ (for some $a>0$) to be the minimal-surface type of diffusivity (see Section 2.3). Let $u(x, y; t)$ be digitized to $u(i\Delta x, j\Delta y, n\Delta t)$, which for short is simply denoted by $u_n(i, j)$. The differentials are then approximated using numerical schemes inspired by computational fluid dynamics (CFD) [4,20],

$$\frac{u_{n+1}-u_n}{\Delta t} = \Delta_{x-}\left(\frac{\Delta_{x+}u_n}{\sqrt{(\Delta_{x+}u_n)^2+(\Delta_y^m u_n)^2+a^2}}\right) + \Delta_{y-}\left(\frac{\Delta_{y+}u_n}{\sqrt{(\Delta_x^m+u_n)^2+(\Delta_{y+}u_n)^2+a^2}}\right),$$

where, $\Delta_{x+}u=u(i+1,j)-u(i,j)/\Delta x$ and $\Delta_{x-}u=(u(i,j)-u(i-1,j)/\Delta x)$ denote the forward and backward finite differences,

$$\Delta_x^m u = \frac{1}{\Delta x}\min\bmod\,(u(i+1,j)-u(i,j), u(i,j)-u(i-1,j)),$$

and

$$\min\bmod(a,b) = \frac{\operatorname{sign}(a)+\operatorname{sign}(b)}{2}\min(|a|,|b|).$$

Color images can be treated either as RGB 3D vectorial functions [7], or as tensor products of different color components as in CB and HSV nonlinear color models (see, e.g. Chan, Kang, and Shen [8]). Consequently, the Perona–Malik baking step can be applied either directly to the RGB vectors or individually to the nonlinear color components. The latter has been carried out in great details in the work of Chan, Kang, and Shen [8].

For the shake step, we use the Newton–Raphson method (3) with a simple difference scheme. Let I denote the digitized lattice approximating the real line $\mathbb{R}$ (with the same step size Δx as in the bake step). For each $y\in(0, H)$,

$$s_{n+1} = \operatorname{round}\left(s_n + \frac{(f_0-f(I_n))\bullet\Delta_+ f(I_n)}{\|\Delta_+ f(I_n)\|^2}\right).$$

here, f_0 denotes the noisy and jittered horizontal profile vector, $f(I_n)$ the vector on the shifted lattice $I_n=I+s_n$ for the baked profile, and Δ_+ the forward finite difference scheme. The $\bullet$ symbol denotes the ordinary Euclidean vector inner product, i.e. $g(I)\bullet h(I)=\sum_{i\in I} g(i)h(i)$. These formulae easily extend to color images.

The bake and shake algorithm is implemented iteratively as follows (also see Fig. 2). The noisy and jittered image u_0 is first *baked* to image u (or denoted by $u_{0.5}$ in Fig. 2), which is then employed as a mollified approximation to the ideal image to estimate the optimal jitter s. The given noisy and jittered image u_0 is then dejittered to $u_1(x, y)=u_0(x-s, y)$. If $s(y)$ faithfully recovers the true jitters $t(y)$, u_1 would become jitter free and its intensity noise n can be cleaned up using any classical denoising schemes such as Rudin–Osher–Fatemi's [4]. In practice, u_1 still contains minor jitters, and the iteration of the

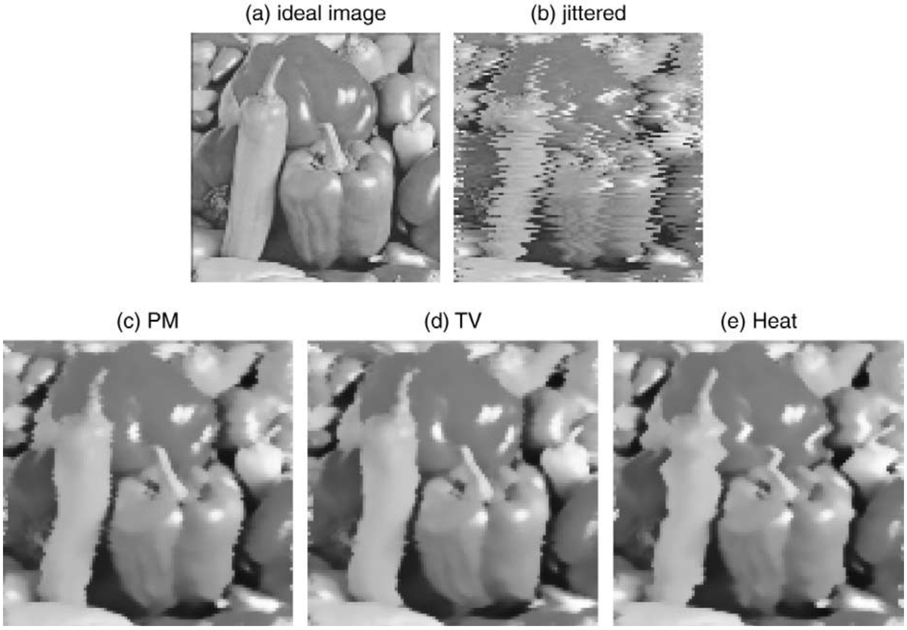

Fig. 7. Comparing the dejittering performance of bake-and-shake algorithms using three different types of the baking tools (for a gray-scale image): (a) the ideal image, (b) the jittered image u_0, (c) the dejittered image using Perona–Malik baking, (d) the dejittered image using TV-restoration baking, (e) the dejittered image using the ordinary linear heat diffusion as the baking step.

above procedure can further reduce the jittering effect. Our numerical experiments show that several loops (up to five in all our examples) of the bake-and-shake algorithm suffice to produce satisfactory results:

$$u_0 \to u_1 \to u_2 \to \cdots \to u_5.$$

3.2. Computational examples

We now show several computational examples to demonstrate the performance of the bake-and-shake algorithm.

In Fig. 7, we first compare the dejittering performances based on three different types of baking mechanisms: (a) the Perona–Malik diffusion proposed in this paper, (b) the linear diffusion model $u_t = \Delta u$, as well as (c) the total variation image restoration model (TV) of Rudin, Osher, and Fatemi [4]:

$$\min_u \left\{ \int_\Omega |\nabla u| \, \mathrm{d}x \, \mathrm{d}y + \frac{\lambda}{2} \int_\Omega (u - u_0)^2 \, \mathrm{d}x \, \mathrm{d}y \right\}.$$

Notice that the TV denoising model has been employed as an intermediate baking or smoothing tool, rather than the dejittering model. Due to the edge-preservation characteristic shared by both the Perona–Malik diffusion and TV restoration, the dejittering performance resulting from these two baking mechanisms are quite comparable, as manifest in Fig. 7. On the other hand, as expected from earlier sections, baking based on the linear heat diffusion results in relatively poor dejittering performance.

In Fig. 8, we apply the above three baking mechanisms to a stand *color* test image. For this example, both the bake and shake steps are applied to the RGB vectorial images. And the computational results again confirm the advantage of the Perona–Malik baking over the linear heat baking for the hake-and-shake algorithm.

Figs. 9 and 10 further highlight the dejittering performance of the bake-and-shake algorithm on RGB color images based on the Perona–Malik baking mechanism. Notice that in Fig. 9, both the jittering and the intensity noise are already quite severe. In both examples, although the bake step has been repeatedly applied, the edges in the final outputs remain remarkably sharp since the bake step has been employed merely as intermediate messengers or information extractors.

4. Conclusion and future work

In this paper, we have invented a two-step bake-and-shake dejittering model for restoring jittered and noisy video images. The bake step extracts valuable positional information of the target ideal image, based upon which the shake step optimally estimates the individual horizontal jitters. By partitioning the dejittering task into two separate key components—bake and shake, we introduce the vast degree of freedom in choosing different mechanisms for the baking and shaking steps.

Both the theoretical analysis and numerical results confirm the outstanding advantage of baking on Perona–Malik diffusions over linear heat diffusions. Combined with the least-square minimization using the Newton–Raphson iterative scheme, Perona–Malik baking eventually leads to a robust and efficient shaking for optimal estimation of random horizontal jitters.

Our future work will focus on more complex image structure that are in compatible wit the jittering or the baking

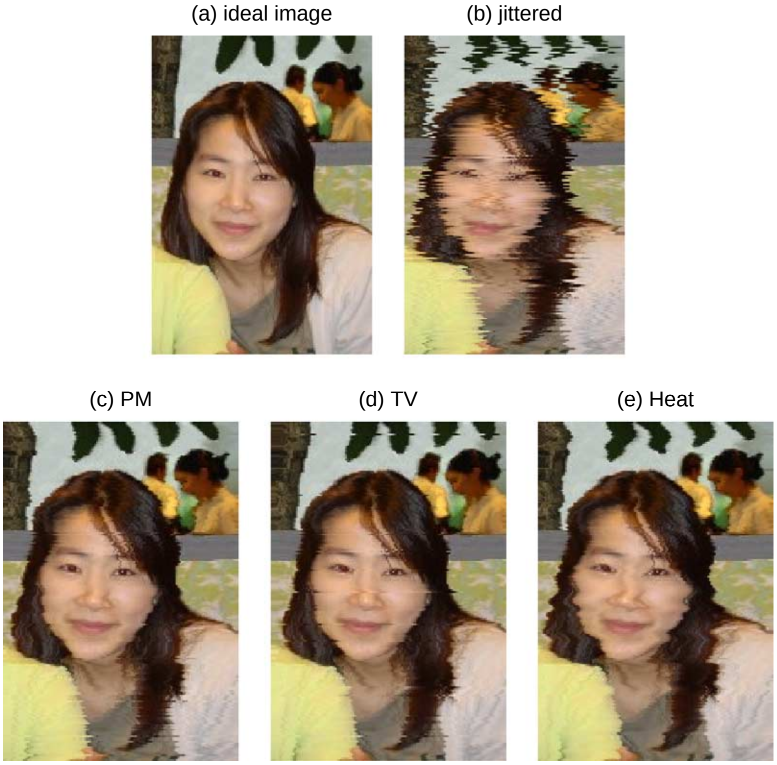

Fig. 8. Comparing dejittering performances based on three different baking tools as applied to an RGB vectorial test image: (a) the ideal image, (b) the jittered image u_0, (c) bake and shake by Perona–Malik baking, (d) bake and shake by TV-restoration baking, and (e) bake and shake by the linear heat baking.

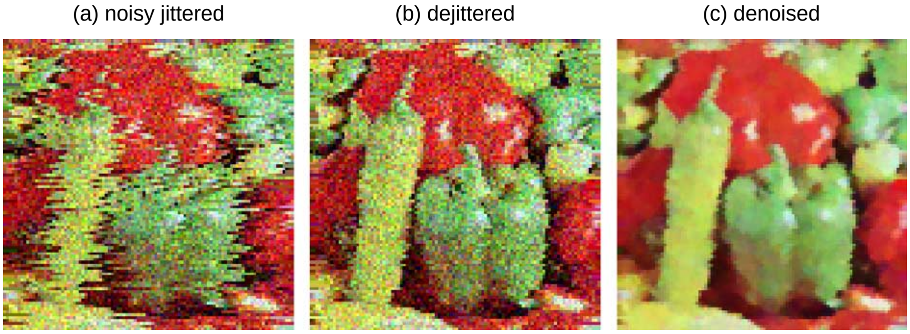

Fig. 9. (a) The noisy jittered image u_0, (b) direct output from the iterated bake-and-shake algorithm (see Fig. 2), and (c) when the output is further denoised.

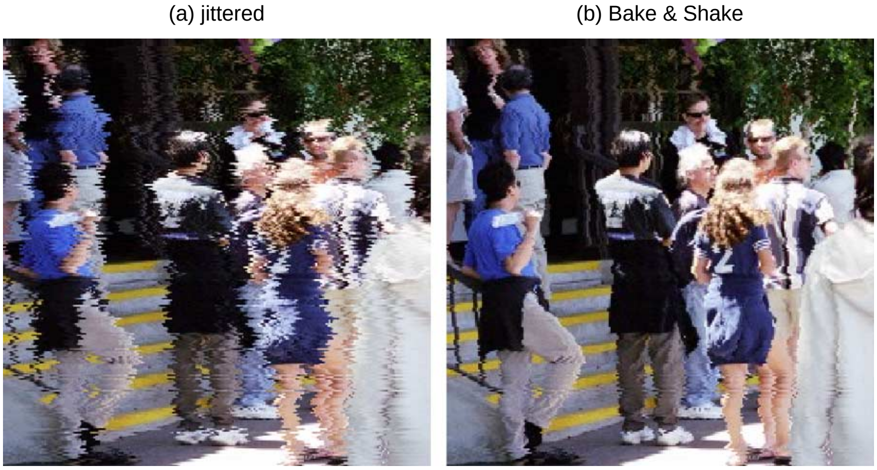

Fig. 10. (a) A jittered test image, and (b) the dejittered image by bake and shake.

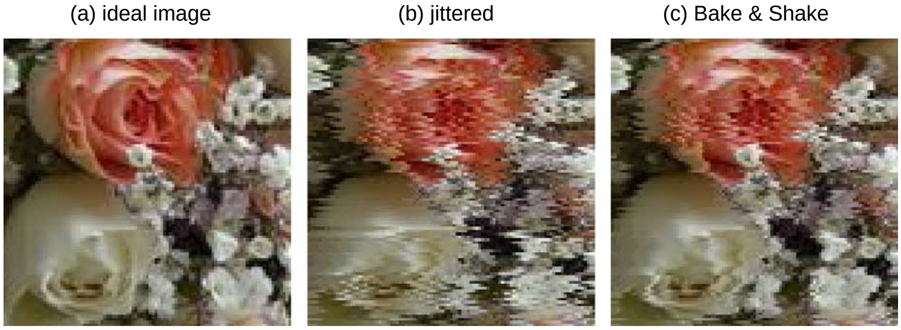

Fig. 11. Insufficiency of the bake-and-shake algorithm for dijittering pure texture images: (a) an ideal texture image (roses), (b) the jittered image u_0, (c) bake and shake via Perona–Malik. Since Perona–Malik diffusions are mainly designed for blocky edges and perform less ideally for statistical textures, the resulting bake-and-shake algorithm becomes less effective for image dejittering (see the remanent jittering effect in (c)).

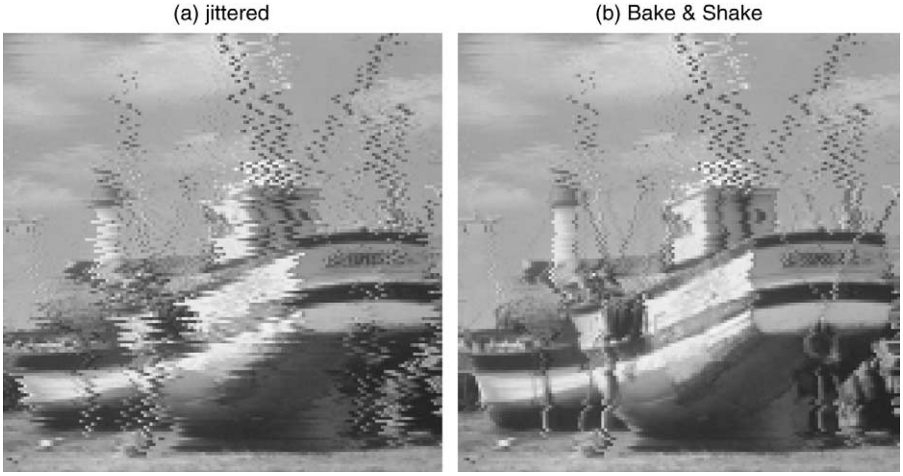

Fig. 12. (a) The noisy jittered image u_0, and (b) the output of the bake-and-shake model. Notice how the jittered thin sail spars in the upper portion fail to be dejittered. After random jittering, thin line (or 1D) structures merely become scattered points, and any *diffusion based* baking process (whether linear or nonlinear) cannot deal with such isolated points.

processes. More specifically, we shall study how to dejitter (a) image without clear edge information such as generic textures in natural image, and (b) images with thin line structures, which after the jittering process often only show collections of *isolated* points rather than 2D homogeneous regions of similar pixels. Figs. 11 and 12 and their captions further clarify these two points separately.

Acknowledgements

This work has been partially supported by the NSF (USA) under grant numbers DMS-0312223 and DMS-0202565.

References

[1] A. Kokaram, P. Rayner. An algorithm for line registration of TV images based on a 2-D AR model, Signal Processing VI, Theories and Applications, 1992, pp. 1283–1286.

[2] A. Kokaram, P.M.B. Roosmalen, P. Rayner, J. Biemond, Line registration of jittered video, IEEE International Conference on Acoustics, Speech, and Signal Processing, 1997, pp. 2553–2556.

[3] J. Shen, Bayesian video dejittering by BV image model, SIAM J. Appl. Math. 64 (5) (2004) 1691–1708.

[4] L. Rudin, S. Osher, E. Fatemi, Nonlinear total variation based noise removal algorithms, Physica D 60 (1992) 259–268.

[5] P. Perona, J. Malik, Scale-space and edge detection using anisotropic diffusion, IEEE Trans. Pattern Anal. Mach. Intell. 12 (1990) 629–639.

[6] J. Stoer, R. Bulirsch, Introduction to Numerical Analysis, Springer, New York, 1992.

[7] P.V. Blomgren, T.F. Chan, Color TV: total variation methods for restoration of vector valued images, IEEE Trans. Image Process. 7 (1998) 304–309.

[8] T.F. Chan, S.H. Kang, J. Shen, Total variation denoising and enhancement of color images based on the CB and HSV color models, J. Visual Commun. Image Rep. 12 (4) (2001) 22–435.

[9] R. Kimmel, N. Sochen, Orientation diffusion or how to comb a porcupine?, J. Visual Commun. Image Rep. 13 (2001) 238–248.

[10] P. Perona, Orientation diffusion, IEEE Trans. Image Process. 7 (3) (1998) 457–467.

[11] P.E. Trahanias, D. Karako, A.N. Venetsanopoulos, Directional processing of color images: theory and experimental results, IEEE Trans. Image Process. 5 (6) (1996) 868–880.

[12] B. Tang, G. Sapiro, V. Caselles, Color image enhancement via chromaticity diffusion, IEEE Trans. Image Process. 10 (2001) 701–707.

[13] T.M. Cover, J.A. Thomas, Elements of Information Theory, Wiley, New York, 1991.

[14] A. Chambolle, P.L. Lions, Image recovery via total variational minimization and related problems, Numer. Math. 76 (1997) 167–188.

[15] T.F. Chan, S. Osher, J. Shen, The digital TV filter and non-linear denoising, IEEE Trans. Image Process. 10 (2) (2001) 231–241.

[16] A. Kokaram, Motion Picture Restoration, Springer, London, 1998.

[17] A. Witkin, Scale space filtering—a new approach to multi-scale description, in: R. Ullmann (Ed.), Image Understanding, Ablex, New Yersey, USA, 1984, pp. 79–95.

[18] J. Weickert, Anisotropic Diffusion in Image Processing, Teubner-Verlag, Stuttgart, Germany, 1998.

[19] F. Catté, P.L. Lions, J.M. Morel, T. Coll, Image selective smoothing and edge detection by nonlinear diffusion, SIAM J. Numer. Anal. 29 (1992) 182–193.

[20] T.F. Chan, S.H. Kang, J. Shen, Euler's elastica and curvature based inpainting, SIAM J. Appl. Math. 63 (2) (2002) 564–592.