James–Stein Type Center Pixel Weights for Non-Local Means Image Denoising

Yue Wu, Brian Tracey, Premkumar Natarajan, and Joseph P. Noonan

***Abstract*—Non-Local Means (NLM) and its variants have proven to be effective and robust in many image denoising tasks. In this letter, we study approaches to selecting center pixel weights (CPW) in NLM. Our key contributions are 1) we give a novel formulation of the CPW problem from a statistical shrinkage perspective; 2) we construct the James–Stein shrinkage estimator in the CPW context; and 3) we propose a new local James–Stein type CPW (LJSCPW) that is locally tuned for each image pixel. Our experimental results showed that compared to existing CPW solutions, the LJSCPW is more robust and effective under various noise levels. In particular, the NLM with the LJSCPW attains higher means with smaller variances in terms of the peak signal and noise ratio (PSNR) and structural similarity (SSIM), implying it improves the NLM denoising performance and makes the denoising less sensitive to parameter changes.**

***Index Terms*—Adaptive algorithm, image denoising, James–Stein estimator, non-local means, shrinkage estimator.**

I. Introduction

IMAGE noise commonly exists in the image acquisition, quantization, transmission and many other processing stages. A digital image contaminated by noises leads to visible loss in image quality and can impact many advanced image processing and computer vision tasks such as tracking, recognition, classification, etc. The importance of image denoising is therefore widely recognized.

Conventional image denoising methods such as moving average filters, Wiener filters, and wavelet filter banks are strongly related to standard filtering [1]. These filter-based image denoising techniques are commonly of low complexity and can be easily implemented. However, their performance is not always adequate. With the increased computational capacity of modern processors, many advanced denoising techniques are now feasible. Among these techniques, the Non-Local Means (NLM) method [1], [2] has attracted significant attention in recent years. The NLM denoises an image pixel as the weighted sum of its noisy neighbors, where each weight reflects the similarity between the local patch centered at the pixel to be denoised and the patch centered at the neighbor pixel. In this way, NLM adapts the denoising process for each pixel and thus outperforms conventional techniques [1]. Other state-of-the-art denoising approaches include the BM3D algorithm [3], which is compared to NLM in [4], [5].

Many improvements on the original NLM have been proposed in recent years. These discussions mainly focus on three questions: 1) how to pick NLM parameters heuristically or automatically [6], [7]; 2) how to accelerate NLM computations [8]–[10] and 3) how to adjust the NLM framework to achieve better performance [4], [5], [11]–[14].

The importance of CPW in NLM has long been known [1], [7], and various weights have been designed [7]. However, the methods proposed are non-ideal as they do not consider all aspects of CPW problem (see details in Section III.A). Thus, new CPWs need to be designed. In this letter, we discuss the CPW problem in NLM and propose new solutions based on the James–Stein estimator [15]. The rest of the letter is organized as: Section II reviews the NLM and related work on CPWs; Section III discusses the new formulation of the CPW problem and the new James–Stein type CPWs; Section IV shows experimental comparisons; and Section V concludes the letter.

II. A Brief Review

A. The Classic Non-Local Means Algorithm

Let $\mathbf{x} = \{x_l\}_{l\in\mathbb{I}}$ be a 2D image defined on the spatial domain $\mathbb{I}$, with $l = (l_1, l_2)$ the lth pixel located at the intersection of the l_1th row and the l_2th column. Pixels of the observed noisy image $\mathbf{y} = \{y_l\}_{l\in\mathbb{I}}$, are assumed to be contaminated by an i.i.d. zero-mean Gaussian noise with a variance of σ^2, namely

$$y_l = x_l + n_l, \text{ and } n_l \sim \mathcal{N}(0, \sigma^2). \tag{1}$$

The classic NLM method [1], [2] estimates the clean pixel x_l by using all pixels within a prescribed search region $\mathbb{S}$, typically a square or a rectangular region. Specifically, the estimated $\hat{x}_l$ is the weighted sum of y_l's noisy neighbors as

$$\hat{x}_l = \sum_{k\in\mathbb{S}} \frac{w_{l,k} y_k}{\sum_{k\in\mathbb{S}} w_{l,k}} \tag{2}$$

where each weight is computed by quantifying the similarity between two local patches around noisy pixels y_l and y_k as shown in (3). Here, G_α is a Gaussian weakly smooth kernel [1] and $\mathbb{P}$ denotes the local patch, typically a square centered at the pixel and h is a temperature parameter controlling the behavior of the weight function.

$$w_{l,k} = \exp(-\sum_{j\in\mathbb{P}} G_\alpha (y_{l+j} - y_{k+j})^2 / 2h) \tag{3}$$

Manuscript received November 12, 2012; revised January 26, 2013; accepted January 27, 2013. Date of publication February 18, 2013; date of current version March 07, 2013. This work was supported by the Tufts subcontract to BBN on MADCAT contract (Prime Contract HR0011-08-C-004). The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Giuseppe Scarpa.

Y. Wu, B. Tracey, and J. P. Noonan are with the Department of Electrical and Computer Engineering, Tufts University, Medford, MA 02155 USA (e-mail: ywu03@ece.tufts.edu).

P. Natarajan is with the Raytheon BBN Technologies, Cambridge, MA 02138 USA.

Digital Object Identifier 10.1109/LSP.2013.2247755

B. Existing Central Pixel Weights

The CPW in the classic NLM is unitary, because (3) implies $w_{l,l} = 1$ for all $l \in \mathbb{I}$. However, this unitary CPW is reported not to perform well in many cases [7]. Indeed, in the case that y_l is highly noisy and y_l has a rare patch (implying that for all $k \in \mathbb{S}\backslash\{l\}$ the non-center weights $w_{l,k} \ll 1$), a unitary CPW means the contribution of the noisy pixel y_l dominates in the denoised pixel $\hat{x}_l$, implying poor performance.

In addition to this CPW, several other CPWs have been proposed and used in the NLM community to enhance performance. These include the zero CPW (5), the Stein CPW (6), and the max CPW (7). In the rest of the letter, we use v_l to denote these existing CPWs.

$$v_l^{\text{one}} = 1 \tag{4}$$
$$v_l^{\text{zero}} = 0 \tag{5}$$
$$v_l^{\text{stein}} = \exp(-\sigma^2|\mathbb{P}|/h) \tag{6}$$
$$v_l^{\max} = \max_{k\in\mathbb{S}\backslash\{l\}}(w_{l,k}) \tag{7}$$

These CPWs are of two groups: global CPWs (4), (5) and (6)) and local CPWs (7). The global CPWs use a constant CPW for all pixels, while the local CPWs vary for pixels. In the next section, we will show that all of them fail to take all variables into full consideration and therefore oversimplify the CPW problem.

III. Shrinkage Based Center Pixel Weights

A. The CPW Problem in the Form of Shrinkage Estimator

To fully reveal the CPW problem, we separate the contributions of the non-center and of the center pixels in the NLM denoised pixel $\hat{x}_l$ (2)

$$\hat{x}_l = \frac{W_l}{W_l + v_l}\hat{z}_l + \frac{v_l}{W_l + v_l}y_l \tag{8}$$

where W_l is the summation of all non-center weights

$$W_l = \sum_{k\in\mathbb{S}\backslash\{l\}} w_{l,k} \tag{9}$$

and $\hat{z}_l$ is the denoised pixel by using all non-center weights.

$$\hat{z}_l = \sum_{k\in\mathbb{S}\backslash\{l\}} w_{l,k}y_k/W_l. \tag{10}$$

If we are given an optimal $\hat{x}_l$ and solve for v_l, we see that the optimal v_l is a function of $W_l, \hat{z}_l$ and y_l. Thus, a CPW v_l that does not consider all these variables is incomplete. We note that the global CPWs $v_l^{\text{one}} v_l^{\text{zero}}$ and v_l^{stein} neglect all three, while the local CPW $v_l^{\max}$ neglects y_l.

Let p_l be the fraction ($p_l \in [0,1]$) of the contribution of the center pixel y_l in $\hat{x}_l$, namely

$$p_l = v_l/(v_l + W_l). \tag{11}$$

p_l is then a normalized version of v_l. Consequently, the NLM-CPW problem in (8) can be rewritten as

$$\hat{x}_l = (1 - p_l)\hat{z}_l + p_l y_l \tag{12}$$

and is a so-called shrinkage estimator, which improves an existing estimator by using the raw data. In the context of the NLM, the existing estimator is $\hat{z}_l$ and the raw data is the noisy pixel y_l. The effect of the CPW is to tune the final denoised pixel $\hat{x}_l$ somewhere between $\hat{z}_l$ and y_l, or equivalently to shrink y_l towards to $\hat{z}_l$.

B. The James–Stein Center Pixel Weight

One important result in shrinkage estimators is the James–Stein estimator [15]. It states that for an unknown parameter vector $\mathbf{a}$ and observations of $\mathbf{b}$ with the relation,

$$\mathbf{b} \sim \mathcal{N}(\mathbf{a}, \sigma^2\mathbf{I}) \tag{13}$$

there exists a James–Stein estimator that shrinks towards an arbitrary vector $\mathbf{c}$ in the form that

$$\hat{\mathbf{a}}^{\text{JS}} = \mathbf{c} + q(\mathbf{b} - \mathbf{c}) = (1 - q)\mathbf{c} + q\mathbf{b} \tag{14}$$

with the coefficient q of form (15) [15].

$$q = 1 - (m - 2)\sigma^2/\|\mathbf{b} - \mathbf{c}\|^2 \tag{15}$$

The James–Stein estimator is a classic solution that minimizes the risk of estimation in terms of the error $E[\|\mathbf{a} - \hat{\mathbf{a}}\|^2]$ [16], where $\|\cdot\|$ denotes the L^2-norm.

In the context of NLM-CPW problem, the James–Stein (JS) based CPW has the weight of form (16),

$$p^{\text{JS}} = 1 - (m - 2)\sigma^2/\|\mathbf{y} - \hat{\mathbf{z}}\|^2 \tag{16}$$

where $m = |\mathbb{I}|$ is the number of pixels in the image, and the corresponding new estimator is

$$\hat{\mathbf{x}}^{\text{JS}} = (1 - p^{\text{JS}})\hat{\mathbf{z}} + p^{\text{JS}}\mathbf{y}. \tag{17}$$

C. Local Adapted James–Stein Center Pixel Weights

Although the JSCPW considers all $W_l, \hat{x}_l$ and y_l, it is still a global CPW and gives an identical weight to all pixels. However, the denoising process is always biased rather than unbiased for each pixel. Thus, ideally we want a locally adapted CPW p_l for each pixel. One natural idea is to replace $\|\mathbf{y} - \hat{\mathbf{z}}\|^2$ in (16) with $\|y_l - \hat{z}_l\|^2$, but this does not lead to a stable solution, because of the inaccurate point-wise estimation. Alternatively, we can view each image block as a small image and thus the JSCPW (16) computed for a local block gives a local CPW adapted to each pixel.

Without loss of generality, let $\widehat{\mathbf{zb}}_l = \{\widehat{z_{l+b}}|\forall b \in \mathbb{B}\}$ and $\mathbf{yb}_l = \{y_{l+b}|\forall b \in \mathbb{B}\}$ be two local image blocks around the lth pixel in $\hat{\mathbf{z}}$ and $\mathbf{y}$, respectively. Given a prescribed local block region $\mathbb{B}$, the local James–Stein (LJS) CPW can be found as

$$p_l^{\text{LJS}} = 1 - (|\mathbb{B}| - 2)\sigma^2/\|\mathbf{yb}_l - \hat{\mathbf{zb}}_l\|^2. \tag{18}$$

In this way, we construct a local CPW at each pixel, and thus the denoised pixel by using LJSCPW can be written as

$$\hat{x}_l^{\text{LJS}} = \left(1 - p_l^{\text{LJS}}\right)\hat{z}_l + p_l^{\text{LJS}}y_l. \tag{19}$$

TABLE I
PSNR/SSIM COMPARISONS (MEAN± STANDARD DEVIATION) FOR VARIOUS CENTER PIXEL WEIGHTING SCHEMES

Img		3×3 Patch					5×5 Patch				
	σ	v_l^{zero}	v_l^{one}	v_l^{stein}	v_l^{max}	p_l^{LJS}	v_l^{zero}	v_l^{one}	v_l^{stein}	v_l^{max}	p_l^{LJS}
cameraman	**10**	29.94 ± 1.92/.87 ± .04	31.97 ± 1.01/.88 ± .05	30.69 ± 2.11/.87 ± .04	32.01 ± 0.87/.88 ± .03	32.79 ± 0.67/.90 ± .04	27.41 ± 2.45/.84 ± .03	31.20 ± 1.09/.86 ± .05	28.53 ± 2.78/.85 ± .03	31.26 ± 1.03/.87 ± .02	32.85 ± 0.41/.91 ± .02
	20	27.70 ± 0.96/.78 ± .07	27.85 ± 1.31/.78 ± .09	27.85 ± 1.03/.78 ± .07	27.98 ± 0.98/.78 ± .07	28.64 ± 0.77/.80 ± .07	26.34 ± 1.21/.78 ± .04	27.08 ± 1.62/.78 ± .09	26.78 ± 1.46/.79 ± .04	27.29 ± 1.49/.79 ± .04	28.77 ± 0.52/.83 ± .04
	40	23.91 ± 1.09/.63 ± .11	23.73 ± 1.58/.62 ± .12	23.88 ± 1.19/.63 ± .11	23.93 ± 1.09/.63 ± .11	24.51 ± 1.01/.64 ± .11	23.15 ± 1.28/.68 ± .06	22.86 ± 1.80/.66 ± .11	23.19 ± 1.31/.68 ± .07	23.24 ± 1.34/.68 ± .06	24.60 ± 0.67/.70 ± .07
	60	21.45 ± 1.18/.50 ± .11	21.24 ± 1.72/.49 ± .12	21.39 ± 1.31/.50 ± .11	21.45 ± 1.18/.50 ± .11	21.86 ± 1.17/.51 ± .11	21.33 ± 1.14/.60 ± .08	20.91 ± 1.94/.58 ± .12	21.31 ± 1.17/.59 ± .08	21.35 ± 1.14/.60 ± .08	22.35 ± 0.74/.60 ± .08
barbara	**10**	31.87 ± 0.94/.90 ± .02	31.91 ± 1.12/.89 ± .04	32.04 ± 0.96/.90 ± .02	32.10 ± 0.90/.90 ± .02	32.57 ± 0.59/.91 ± .02	30.77 ± 1.45/.89 ± .02	31.32 ± 1.36/.88 ± .04	31.23 ± 1.61/.89 ± .02	31.58 ± 1.31/.89 ± .02	32.83 ± 0.53/.91 ± .01
	20	27.65 ± 1.02/.79 ± .05	27.41 ± 1.37/.78 ± .07	27.61 ± 1.06/.79 ± .05	27.66 ± 1.01/.79 ± .05	28.11 ± 0.75/.80 ± .05	27.22 ± 1.51/.79 ± .04	26.84 ± 1.71/.77 ± .08	27.27 ± 1.52/.79 ± .04	27.29 ± 1.54/.79 ± .04	28.45 ± 0.82/.82 ± .03
	40	23.37 ± 1.02/.60 ± .07	23.15 ± 1.52/.59 ± .09	23.31 ± 1.13/.60 ± .07	23.38 ± 1.02/.60 ± .07	23.66 ± 0.93/.61 ± .07	23.44 ± 1.27/.64 ± .05	22.96 ± 1.82/.62 ± .09	23.40 ± 1.27/.64 ± .05	23.45 ± 1.26/.64 ± .05	24.16 ± 0.85/.67 ± .04
	60	21.26 ± 1.14/.47 ± .08	21.05 ± 1.69/.47 ± .09	21.20 ± 1.28/.47 ± .08	21.26 ± 1.14/.47 ± .08	21.40 ± 1.15/.48 ± .08	21.60 ± 0.98/.54 ± .05	21.11 ± 1.93/.52 ± .09	21.55 ± 1.04/.54 ± .06	21.61 ± 0.97/.54 ± .05	21.99 ± 0.78/.55 ± .05
boat	**10**	31.07 ± 1.14/.84 ± .02	31.38 ± 1.07/.84 ± .03	31.31 ± 1.20/.84 ± .03	31.51 ± 0.95/.84 ± .02	32.13 ± 0.48/.86 ± .02	29.47 ± 1.69/.79 ± .04	30.43 ± 1.23/.80 ± .04	29.97 ± 1.88/.80 ± .04	30.57 ± 1.32/.81 ± .04	32.10 ± 0.42/.85 ± .01
	20	27.75 ± 0.92/.73 ± .04	27.59 ± 1.30/.72 ± .06	27.75 ± 0.98/.73 ± .05	27.79 ± 0.93/.73 ± .04	28.25 ± 0.70/.74 ± .04	26.81 ± 1.26/.70 ± .04	26.61 ± 1.51/.69 ± .07	26.91 ± 1.31/.70 ± .04	26.94 ± 1.33/.71 ± .04	28.18 ± 0.59/.75 ± .02
	40	24.05 ± 1.07/.57 ± .08	23.84 ± 1.62/.56 ± .09	24.00 ± 1.20/.57 ± .08	24.06 ± 1.07/.57 ± .08	24.36 ± 1.04/.58 ± .08	23.84 ± 1.06/.59 ± .04	23.41 ± 1.82/.57 ± .09	23.82 ± 1.09/.59 ± .05	23.86 ± 1.07/.59 ± .04	24.62 ± 0.68/.61 ± .04
	60	22.00 ± 1.29/.46 ± .09	21.78 ± 1.88/.45 ± .10	21.93 ± 1.45/.46 ± .10	22.00 ± 1.30/.46 ± .09	22.16 ± 1.32/.46 ± .09	22.31 ± 0.89/.52 ± .06	21.82 ± 2.06/.50 ± .10	22.26 ± 1.00/.52 ± .06	22.31 ± 0.89/.52 ± .06	22.75 ± 0.73/.52 ± .06
couple	**10**	30.81 ± 1.11/.85 ± .02	31.12 ± 1.12/.85 ± .03	31.04 ± 1.18/.85 ± .03	31.23 ± 1.05/.85 ± .02	31.89 ± 0.48/.87 ± .02	29.03 ± 1.67/.80 ± .04	30.14 ± 1.40/.81 ± .04	29.53 ± 1.91/.81 ± .04	30.28 ± 1.55/.81 ± .04	31.88 ± 0.49/.86 ± .01
	20	27.16 ± 0.95/.73 ± .04	26.99 ± 1.25/.72 ± .06	27.15 ± 0.99/.72 ± .04	27.19 ± 0.95/.73 ± .04	27.69 ± 0.65/.74 ± .04	26.20 ± 1.34/.69 ± .05	26.01 ± 1.50/.68 ± .07	26.30 ± 1.40/.69 ± .05	26.34 ± 1.43/.69 ± .05	27.67 ± 0.64/.74 ± .02
	40	23.59 ± 0.98/.55 ± .06	23.38 ± 1.51/.54 ± .08	23.53 ± 1.10/.55 ± .07	23.59 ± 0.98/.55 ± .06	23.83 ± 0.93/.56 ± .07	23.37 ± 0.98/.56 ± .04	22.95 ± 1.69/.54 ± .08	23.35 ± 1.00/.56 ± .04	23.38 ± 0.98/.56 ± .04	24.06 ± 0.62/.59 ± .04
	60	21.76 ± 1.24/.44 ± .08	21.55 ± 1.81/.43 ± .09	21.69 ± 1.39/.44 ± .08	21.76 ± 1.24/.44 ± .08	21.87 ± 1.26/.44 ± .08	22.00 ± 0.80/.49 ± .05	21.53 ± 1.97/.47 ± .09	21.95 ± 0.91/.48 ± .05	22.01 ± 0.80/.49 ± .05	22.35 ± 0.66/.49 ± .05
fingerprint	**10**	30.72 ± 0.73/.95 ± .01	30.73 ± 0.96/.95 ± .01	30.93 ± 0.69/.95 ± .01	30.90 ± 0.73/.95 ± .01	31.17 ± 0.56/.96 ± .01	28.18 ± 1.10/.91 ± .02	29.13 ± 1.08/.92 ± .02	28.81 ± 1.24/.92 ± .02	29.12 ± 1.14/.92 ± .02	30.60 ± 0.38/.95 ± .00
	20	26.75 ± 1.00/.89 ± .02	26.44 ± 1.34/.88 ± .03	26.70 ± 1.02/.89 ± .02	26.76 ± 0.99/.89 ± .02	26.99 ± 0.79/.90 ± .02	25.29 ± 1.33/.83 ± .04	25.00 ± 1.44/.83 ± .04	25.39 ± 1.36/.84 ± .05	25.37 ± 1.36/.84 ± .05	26.38 ± 0.70/.88 ± .01
	40	21.64 ± 1.22/.71 ± .06	21.39 ± 1.50/.70 ± .07	21.58 ± 1.25/.71 ± .06	21.65 ± 1.21/.71 ± .06	21.88 ± 1.00/.72 ± .04	21.05 ± 1.77/.64 ± .11	20.57 ± 1.83/.63 ± .10	21.02 ± 1.73/.64 ± .11	21.06 ± 1.77/.64 ± .11	21.82 ± 1.19/.71 ± .06
	60	19.05 ± 1.00/.54 ± .07	18.84 ± 1.39/.53 ± .08	19.00 ± 1.07/.54 ± .07	19.06 ± 0.99/.54 ± .07	19.22 ± 0.88/.56 ± .06	18.80 ± 1.40/.48 ± .13	18.33 ± 1.68/.46 ± .11	18.76 ± 1.37/.48 ± .12	18.81 ± 1.40/.48 ± .13	19.33 ± 1.03/.55 ± .07
hill	**10**	31.35 ± 0.91/.83 ± .02	31.36 ± 1.06/.83 ± .03	31.45 ± 0.95/.83 ± .03	31.50 ± 0.91/.83 ± .02	32.13 ± 0.47/.85 ± .02	29.89 ± 1.23/.78 ± .04	30.28 ± 1.27/.78 ± .05	30.19 ± 1.38/.78 ± .04	30.42 ± 1.31/.79 ± .04	32.05 ± 0.41/.85 ± .01
	20	27.86 ± 0.92/.71 ± .04	27.65 ± 1.32/.70 ± .06	27.82 ± 0.98/.71 ± .05	27.87 ± 0.91/.71 ± .04	28.27 ± 0.72/.72 ± .04	27.18 ± 1.18/.68 ± .04	26.85 ± 1.50/.66 ± .07	27.21 ± 1.20/.68 ± .04	27.23 ± 1.20/.68 ± .04	28.27 ± 0.61/.72 ± .02
	40	24.37 ± 1.09/.54 ± .07	24.15 ± 1.67/.53 ± .09	24.31 ± 1.24/.54 ± .08	24.37 ± 1.09/.54 ± .07	24.55 ± 1.09/.54 ± .08	24.47 ± 0.92/.56 ± .04	24.01 ± 1.86/.54 ± .09	24.43 ± 0.98/.56 ± .05	24.48 ± 0.92/.56 ± .04	24.94 ± 0.69/.57 ± .04
	60	22.59 ± 1.48/.43 ± .09	22.38 ± 2.04/.43 ± .10	22.52 ± 1.63/.43 ± .10	22.59 ± 1.48/.43 ± .09	22.66 ± 1.50/.43 ± .09	23.13 ± 0.80/.49 ± .06	22.63 ± 2.17/.48 ± .10	23.07 ± 0.97/.49 ± .06	23.13 ± 0.80/.49 ± .06	23.32 ± 0.75/.49 ± .05

Intuitively, this LJSCPW helps eliminate the influence of remote image pixels and tunes the optimization locally.

D. Implementation

To lower the computational cost of the LJSCPW in NLM, we construct the integral image [8] $\mathbf{R}$ with 2 operations/pixel for the pixel-wise mean square error between $\hat{\mathbf{z}}$ and $\mathbf{y}$. Each pixel is the summation of form (20).

$$R_l = \sum_{i=\{(i_1,i_2)|i_1\in[1,l_1],i_2\in[1,l_2]\}} (y_i - \hat{z}_i)^2 \tag{20}$$

This integral image $\mathbf{R}$ then allows computation of $\|\mathbf{y}\mathbf{b}_l - \hat{\mathbf{z}}\mathbf{b}_l\|^2$ for an arbitrary rectangular $\mathbb{B}$ with 3 operations/pixel. Considering the extra 4 operations/pixel to compute p_l^{LJS} and the 4 operations/pixel to compute $\hat{x}_l^{\text{LJS}}$, in total the proposed LJSCPW requires additional 13 operations/pixel. Compared to the NLM complexity of $|\mathbb{P}| \times |\mathbb{S}|$ operations/pixel, this shrinkage cost is negligible.

In processing, one may see $p_l^{\text{LJS}} < 0$ for some pixels, which conflicts with our assumption that $p_l^{\text{LJS}} \in [0,1]$. This occurs when $\hat{\mathbf{z}}\mathbf{r}_l$ is a slightly denoised version of $\mathbf{y}\mathbf{r}_l$. It is reasonable to use $\hat{z}_l$ rather than y_l, implying $p_l^{\text{LJS}} = 0$. Therefore, we use the positive part of p_l^{LJS} in (21), namely

$$p_l^{\text{LJS}} = \max\left(1 - (|\mathbb{B}| - 2)\sigma^2 / \|\mathbf{y}\mathbf{b}_l - \hat{\mathbf{z}}\mathbf{b}_l\|^2, 0\right) \tag{21}$$

Since $p_l^{\text{LJS}} = 1$ would indicate the raw pixel is used, it may prove useful in some applications to limit p_l^{LJS} to a user-defined value less than unity. In this letter, however, we allow the shrinkage operator to operate over the full range.

IV. SIMULATION RESULTS

All following simulations are done under the MATLAB r2010a environment with Intel Core CPU at 2.0 GHz. We compare the performance of existing CPWs with the proposed James–Stein type CPW p_l^{LJS} under the classic NLM framework (only the CPW is changed). In particular, we set the search region $\mathbb{S}$ to 31×31 square, use a 15×15 $\mathbb{B}$ centered on the local pixel, and test performance for 3×3, 5×5 and 7×7 patches $\mathbb{P}$, respectively. Test images[1] are gray-scale with additive Gaussian noises of $\sigma \in \{10, 20, 40, 60\}$. We denoise each test image by using 200 temperature parameters h ranging from 1% to 200% of $\sigma^2|\mathbb{P}|$. The denoising performance is then evaluated by computing its mean and standard deviation in terms of PSNR [11], [12] and SSIM [17].

Experimental results of using 3×3 and 5×5 patches are summarized in Table I. The proposed LJSCPW is the only CPW that consistently improves NLM performance (over the zero CPW) in terms of higher overall performance, regardless of patch sizes, test images and noise levels. This implies that LJSCPW outperforms other comparing CPWs and leads to a better NLM solution. Meanwhile, LJSCPW tends to attain a smaller variance in performance scores, implying that LJSCPW leads to a more efficient estimator than other CPWs. It is noticeable that LJSCPW is less sensitive to patch size. With regards to the execution time, the classic NLM costs $2.667 \pm .427$ s and the extra LJSCPW requires additional $0.0034 \pm .0002$ s for 256×256 grayscale images. Simulation code is available in the MATLAB central*File Exchange* site[2]

To better reveal the NLM performance with various CPWs subject to the change of the temperature parameter, we plot their typical performance scores by using the image *cameraman* and 7×7 patch in Fig. 1. This figure clearly shows that the proposed LJSCPW outperforms other CPWs for each h and that performance curves of LJSCPW decay much slower than other CPWs as h increases. Comparing the best performance points for each CPW, we see that the superiority of LJSCPW decreases as the noise level σ increases. This occurs because as σ increases, the optimal shrinkage coefficient p decreases. However, a smaller p in our shrinkage model (19) implies less difference between $\hat{x}_l^{\text{LJS}}$ and $\hat{z}_l$, where $\hat{z}_l$ is simply the denoising result of the zero CPW. This phenomena is more salient when $\hat{z}_l$ is already a very good estimator (see results of $h/\sigma^2|\mathbb{P}|$ in [25%,75%] in Fig. 1).

[1]are available under the page http://www.cs.tut.fi/~foi/GCF-BM3D/BM3D_images.zip as the date of Dec. 10th, 2012.

[2]see http://www.mathworks.com/matlabcentral/fileexchange/40162.

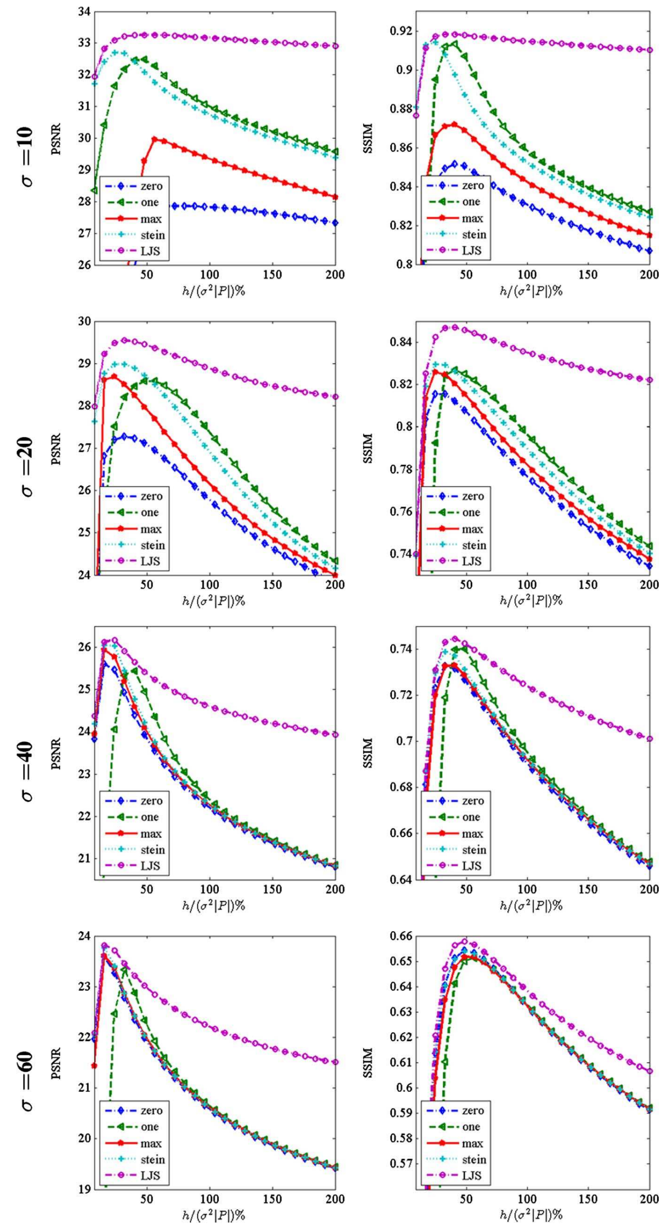

Fig. 1. The PSNR and SSIM performance NLMs with various CPWs.

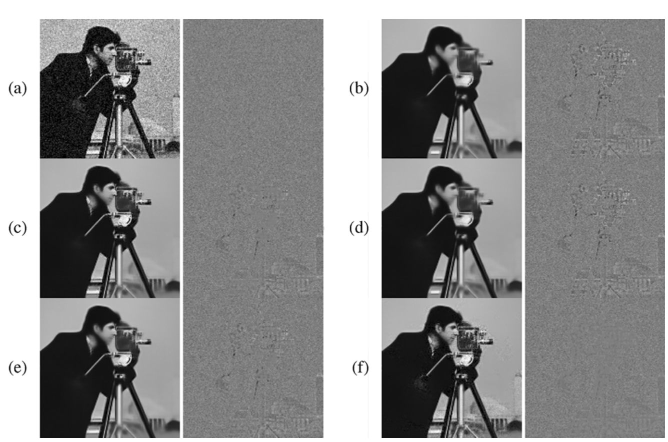

Fig. 2. NLM denoising results and method noises of various CPWs ($h = \sigma^2|\mathbb{P}|$, 7×7 $\mathbb{P}$). (a) Noisy image and true noise $\sigma = 20$ with (PSNR/SSIM) scores 22.14/.41. (b)–(f) NLM denoising results using CPWs. (b) v_l^{zero} with 25.73/.77; (c) v_l^{one} with 27.32/.80; (d) v_l^{max} with 26.12/.78; (e) v_l^{stein} with 26.85/.79; and (f) p_l^{LJS} with 28.68/.84.

However, when $\hat{z}_l$ is not that good, LJSCPW shows noticeable improvement by using noisy data. Finally, we give typical denoising results and method noises of NLM using various CPWs in Fig. 2. It is clear that LJSCPW helps keep image details and weak edges, and has a more random-like method noise than other CPWs.

V. Conclusion

In this letter, we reviewed the CPW problem in NLM and showed that it can be viewed as the well studied statistical shrinkage estimator problem. This formulation opens a new door to the CPW problem: it allows us to use the James–Stein shrinkage estimator and leads us to a new LJSCPW solution. Our experimental results show that the proposed James–Stein type CPW helps NLM to achieve better overall performance, giving higher average PSNR/SSIM scores with a more robust performance in terms of smaller variances. By using this new CPW, NLM is made less sensitive to parameter changes and has a better ability to retain weak edges.

References

[1] A. Buades, B. Coll, and J. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Model. Simul.*, vol. 4, no. 2, pp. 490–530, 2005.

[2] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *IEEE Compu. Soc. Conf. Computer Vision and Pattern Recognition*, Jun. 2005, vol. 2, pp. 60–65, vol. 2.

[3] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, 2007.

[4] C. Deledalle, V. Duval, and J. Salmon, "Non-local methods with shapeadaptive patches (nlm-sap)," *J. Math. Imag. Vis.*, vol. 43, no. 2, pp. 103–120, 2012.

[5] V. Duval, J. Aujol, and Y. Gousseau, "A bias-variance approach for the nonlocal means," *SIAM J. Imag. Sci.*, vol. 4, no. 2, pp. 760–788, 2011.

[6] D. Van De Ville and M. Kocher, "Sure-based non-local means," *IEEE Signal Process. Lett.*, vol. 16, no. 11, pp. 973–976, Nov. 2009.

[7] J. Salmon, "On two parameters for denoising with non-local means," *IEEE Signal Process. Lett.*, vol. 17, no. 3, pp. 269–272, Mar. 2010.

[8] J. Darbon, A. Cunha, T. Chan, S. Osher, and G. Jensen, "Fast nonlocal filtering applied to electron cryomicroscopy," in *IEEE Int. Symp. Biomedical Imaging: From Nano to Macro*, 2008, pp. 1331–1334.

[9] R. Lai and Y. Yang, "Accelerating non-local means algorithm with random projection," *Electron. Lett.*, vol. 47, no. 3, pp. 182–183, 2011, 3.

[10] K. Chaudhury, "Acceleration of the shiftable o(1) algorithm for bilateral filtering and non-local means," *IEEE Trans. Image Process.*, vol. PP, no. 99, 2012, p. 1.

[11] R. Vignesh, B. T. Oh, and C.-C. Kuo, "Fast non-local means (nlm) computation with probabilistic early termination," *IEEE Signal Process. Lett.*, vol. 17, no. 3, pp. 277–280, Mar. 2010.

[12] W. Zeng and X. Lu, "Region-based non-local means algorithm for noise removal," *Electron. Lett.*, vol. 47, no. 20, pp. 1125–1127, 2011, 29.

[13] P. Coupe, J. Manjon, M. Robles, and D. Collins, "Adaptive multiresolution non-local means filter for three-dimensional magnetic resonance image denoising," *IET Image Process.*, vol. 6, no. 5, pp. 558–568, Jul. 2012.

[14] N. Thacker, J. Manjon, and P. Bromiley, "Statistical interpretation of non-local means," *IET Comput. Vis.*, vol. 4, no. 3, pp. 162–172, Sept. 2010.

[15] W. James and C. Stein, "Estimation with quadratic loss," in *Proc. Fourth Berkeley Symp. Mathematical Statistics and Probability*, 1961, vol. 1, no. 36, pp. 1–379.

[16] C. Stein, "Inadmissibility of the usual estimator for the mean of a multivariate normal distribution," in *Proc. Third Berkeley Symp. Mathematical Statistics and Probability*, 1956, vol. 1, no. 399, pp. 197–206.

[17] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004.