A Comparison of Patch-Based Models in Video Denoising

Pablo Arias
CMLA, ENS Cachan
Université Paris-Saclay
94235 Cachan, France
arias@cmla.ens-cachan.fr

Gabriele Facciolo
CMLA, ENS Cachan
Université Paris-Saclay
94235 Cachan, France
facciolo@cmla.ens-cachan.fr

Jean-Michel Morel
CMLA, ENS Cachan
Université Paris-Saclay
94235 Cachan, France
morel@cmla.ens-cachan.fr

***Abstract*—Several state-of-the-art patch-based methods for video denoising rely on grouping similar patches and jointly denoising them. Different models for the groups of patches have been proposed. In general more complex models achieve better results at the expense of a higher running time. But the modeling of the groups of patches is not the only difference between the approaches proposed in the literature. Other differences can be the type of patches, the search strategies used for determining the groups of similar patches and the weights used in the aggregation. This makes it difficult to determine the actual impact of the patch model on the results. In this work we compare two of the models that have produced better results in equal conditions: those assuming sparsity on a fixed transform (like BM3D), against methods that seek to adapt the transform to the group of patches. In addition we propose a third simple model which can be interpreted as a non-local version of the classical DCT denoising and add it to the comparison. We compare the three models with 3D large patches and use the optical flow to guide the search for similar patches, but not to shape the patches. Either one of the three approaches achieves state-of-the-art results, which comes as a consequence of using a large 3D patch size. As expected, the adaptive transform attains better results, but the margin reduces significantly for higher noise levels.**

***Index Terms*—transform domain denoising, Bayesian models, Wiener filter, patch-based methods**

I. Introduction

Patch-based methods are among the state of the art in video denoising. Most of the top performing methods follow the strategy introduced by the BM3D image denoising algorithm [1]: grouping of similar patches, filtering them, and aggregating the filtered patches to form the output video.

To filter the similar patches, BM3D considers a group of similar 2D patches as a 3D signal which is assumed to be sparse on a transformed domain. The transform is fixed, e.g. a separable transform with a DCT on the spatial dimensions and a discrete Haar wavelet on the 3rd dimension. The patches are estimated by applying a shrinkage operator on the transformed domain. Different extensions to video of BM3D have been proposed. V-BM3D [2] is a straightforward extension by considering that similar patches can be in neighboring frames. BM4D and V-BM4D [3], [4] use 3D patches and 4D groups of them. In [5] the authors propose to use 4D groups of shaped adaptive 3D patches to improve the sparsity.

An alternative model for similar patches was proposed [6] for an image denoising algorithm called NL-Bayes. It assumes that the similar patches forming a group are IID samples of a Gaussian distribution. To filter the patches, first this *a priori* Gaussian distribution is estimated and then it is used to compute the MMSE estimates of the patches. This can be interpreted as a shrinkage operator in a transformed domain (the PCA basis). As opposed to BMxD, the transform is optimally adapted to the group of patches. Video NL-Bayes (VNLB) [7] and SPTWO [8] are extensions to video of the NL-Bayes method [6]. The use of an adaptive basis is more costly, but produces results that are significantly better than those obtained with V-BM3D and V-BM4D [7].

However the modeling of the groups of patches is not the only difference between these methods. They differ also in the type of patches they consider: 2D [2], [8], [9], 3D [4], [7], 3D with motion compensation [4]; in the strategies used for to search for similar patches (e.g. [10]) and in the weights used in the aggregation. This makes it difficult to determine the actual impact of the patch model on the results.

For this reason in this work we look at two modifications of the VNLB method which are based on different group filtering strategies for the same $10 \times 10 \times 2$ 3D patch size. The first method, which we call BM4D-OF, can be seen a variant of VNLB that applies the shrinkage filtering proposed in BM4D [4]. In particular, as VNLB, it uses optical flow to aid the patch search (hence the "OF" in BM4D-OF). The second method, called VNLDCT, is another variant of VNLB that uses a fixed DCT basis [11] instead of estimating the optimal basis for each group of patches. The three methods differ on the number of parameters of their underlying models. These parameters have to be estimated together with the patches. For a group of n patches, each of dimension d, VNLB estimates $d(d+1)/2$ parameters, BM4D estimates dn parameters while VNLDCT only estimates $2d$ parameters. Given that the groups of patches have limited size, the estimation task is subject to bias-variance trade-offs: richer models should lead to smaller bias, but are harder to estimate. Table I summarizes the main differences between these models.

Our aim is to compare these models for the groups of similar

Work supported by IDEX Paris-Saclay IDI 2016, ANR-11-IDEX-0003-02, ONR grant N00014-17-1-2552, CNES MISS project, DGA Astrid ANR-17-ASTR-0013-01, DGA ANR-16-DEFA-0004-01, and MENRT.

978-1-5386-0951-4/18/$31.00 ©2018 IEEE

TABLE I
CHARACTERISTICS OF THE COMPARED MODELS

	adaptive base	stack vs patch	mean patch denoising	parameters
VNLB	yes	patch	no	$d(d{+}1)/2$
BM4D-OF	no	stack	yes	dn
NLDCT	no	patch	no	$2d$

patches. The underlying motivation is to identify empirically the best performing model and understand the practical impact (in terms of quality and performance) of the different choices. The contributions of our work are threefold:

(i) We introduce a simple denoising model based on the non-local DCT which serves as a link between video NL-Bayes and the V-BMxD methods.

(ii) We compare in equality of conditions these three models for group filtering, allowing to identify the relative improvement of each strategy and to compare it with its computational cost (using non-optimized CPU code).

(iii) Last but not least, for a more realistic comparison we use test sequences of size 960×540, five times larger than the test videos typically encountered in the related literature

In the next section we review the approaches to be compared. The experimental setup and the results obtained are described in Section III. Section IV collects our conclusions and perspectives.

II. FROM NL-BAYES TO BM4D THROUGH NL-DCT

In this section we describe the three patch-based methods that will be compared in Section III. The three methods follow the same strategy, introduced by [12]: (i) build groups of similar patches, (ii) joint filtering of all patches in the group, (iii) aggregation of the denoised patches in the output image. This procedure is iterated two times. We assume white Gaussian additive noise with known standard deviation σ.

A. Video NL-Bayes

We give here a brief review of the VNLB method introduced in [7]. The method follows an empirical Bayesian approach based on the assumption that groups of similar spatio-temporal 3D patches of the unknown clean video follow a Gaussian *a priori* distribution.

1) Nearest neighbors search: For the three methods compared we use the same strategy for searching for nearest patches, based in [7]. A group of similar patches is built by selecting a patch (the reference patch of the group) and searching for the patches most similar to it within a spatio-temporal search region centered at the reference patch. In [7] the search region consists of a motion-compensated window R of size $w_x \times w_x$ and extending for w_t frames ($w_t/2$ before and $w_t/2$ after the frame of the reference patch). The square spatial slices of the search region are centered following the trajectory of the reference patch, which is defined by following the forward and backward optical flow. Since the optical flow is only used to guide the search, it needs not to be very accurate. We use the TV-L1 method [13] computed with a strong regularization to gain robustness to the noise. The similarity between patches is computed as the squared Euclidean distance. The patches with the n smallest distances (including the reference patch) determine the group.

In the second iteration, the similarity between the patches is computed using the first iterate as the guide, as in [7], [12]. For a more detailes about the nearest neighbor search, we refer the reader to [7].

2) Patch filtering: Let $\boldsymbol{q} \in \mathbb{R}^d$ denote the noisy reference patch of a group, and $\boldsymbol{q}_1 = \boldsymbol{q}, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_n$ be the n nearest neighbors. Here d denotes the dimensionality of the patches. For a patch of size $s_x \times s_x \times s_t$, $d = s_x^2 s_t$. We assume that $\boldsymbol{q}_i = \boldsymbol{p}_i + \boldsymbol{n}_i$, where $\boldsymbol{p}_i$ is the clean patch we wish to estimate $\boldsymbol{p}_i$ and $\boldsymbol{n}_i$ is a vector of additive white Gaussian noise with variance σ^2. Assuming that $\boldsymbol{p}_i \sim \mathcal{N}(\boldsymbol{\mu}, C)$, then if we knew $\boldsymbol{\mu}$ and C, we could estimate $\boldsymbol{p}_i$ as a MAP:

$$\widehat{\boldsymbol{p}}_i = \boldsymbol{\mu} + C(C+\sigma^2 I)^{-1}(\boldsymbol{q}_i - \boldsymbol{\mu}) \Longrightarrow \\ U^T(\widehat{\boldsymbol{p}}_i - \boldsymbol{\mu}) = \Lambda(\Lambda + \sigma^2 I)^{-1} U^T (\boldsymbol{q}_i - \boldsymbol{\mu}). \quad (1)$$

We have used the eigen-decomposition of C, $C = U\Lambda U^T$, where U is the orthogonal matrix of eigenvectors and $\Lambda = \mathrm{Diag}(\lambda_1, \ldots, \lambda_d)$ is the diagonal matrix of eigenvalues.

In practice $\boldsymbol{\mu}$ and C are unknown and have to be estimated from the data. This implies the estimation of d parameters for the mean, and $d(d{+}1)/2$ parameters for the covariance matrix. In the Video NL-Bayes algorithm, they are estimated from the sample mean and sample covariance of the patches $\boldsymbol{q}_1, \ldots, \boldsymbol{q}_n$. In [7] two ways of estimating the sample covariance matrix are proposed. We use the one with the hard-threshold on the eigenvalues.

In the second iteration of the algorithm, the covariance matrix is computed from the patches of the first iterate.

B. Non-local DCT denoising

The Wiener filtering with the estimated patch covariance matrix used in (1) is costly, and scales badly with the size of the patch, since it requires either inverting a matrix $s_x^2 s_t \times s_x^2 s_t$ or computing its (truncated) SVD. It can be viewed as two problems: estimating the d eigenvalues (or variances) Λ and the orthonormal basis U ($d(d{-}1)/2$ parameters). By assuming a fixed basis of principal directions for all patch covariance matrices, we are only left with the problem of estimating $2d$ parameters: the variances Λ and the mean $\boldsymbol{\mu}$.

A natural candidate for the basis U is the DCT basis [14], [15]. This improves the scalability of the algorithm with the patch size, which becomes linear in the dimensionality of the patch due to the separability of the DCT. Thus we propose a variant of the VNLB method [7] by assuming that the principal directions of the Gaussian model are given by the DCT. The resulting algorithm can be interpreted as a non-local version of the classic image denoising [14], [16]–[18].

1) Bayesian DCT shrinkage: Let $U = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_d]$ be a $d \times d$ orthogonal matrix (with $d = s_x^2 s_t$) such that the columns are the DCT basis of a signal of size $s_x \times s_x \times s_t$. The set of similar patches from the noisy image is stored as a $d \times n$ matrix $Q = [\boldsymbol{q}_1, \ldots, \boldsymbol{q}_n]$. The DCT can be computed as $\widetilde{Q} = U^T Q$.

On the first iteration of the algorithm we estimate the mean and variance of each component j from the noisy patches as follows:

$$\hat{\mu}_j = \frac{1}{n}\sum_{i=1}^{n} \widetilde{q}_{ij}, \quad \hat{\lambda}_j = \left(\frac{1}{n}\sum_{j=1}^{n}(\widetilde{q}_{ij} - \mu_j)^2 - \sigma^2\right)_+. \quad (2)$$

Here $(\,\cdot\,)_+$ denotes the positive part, defined for $x \in \mathbb{R}$ by $(x)_+ = \max\{0, x\}$. In the second iteration, in addition to the matrix Q of noisy patches, we have a matrix $G = [\boldsymbol{g}_1, ..., \boldsymbol{g}_n]$ with the corresponding patches from the guide. The mean and variances are computed as the sample mean and sample variance of the patches in G.

In both iterations the MAP estimate for component j of patch i is given by $\widetilde{p}_{ij} = \frac{\hat{\lambda}_j}{\hat{\lambda}_j + \beta\sigma^2}(\widetilde{q}_{ij} - \hat{\mu}_j) + \hat{\mu}_j$, where $\beta > 0$ is a parameter used to control the denoising strength. The final patch estimates are computed by inverting the DCT transform: $P = U\widetilde{P}$. The columns of P are the denoised patches $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n$.

C. BM4D-OF

We include BM4D-OF in our comparison as a representative of the BMxD methods [2]–[4], [12]. The main difference between the BM4D algorithm introduced in [4] and BM4D-OF is that we use the same motion-compensated search region as for NL-Bayes and NL-DCT. This way, the three methods differ only in the filtering of the groups of patches.

BM4D-OF considers the group of 3D video patches as 4D signal (the forth dimension is not a physical dimension, it is determined by the ordering of the patches in the group). The group is filtered by applying a 4D separable transform to the group, shrinking the transformed coefficients and then inverting the transform to reconstruct the group.

We use a 3D DCT as the transform for the spatial and temporal dimensions and a Haar transform for the 4th dimension. In the first iteration, we filter the patch using a hard thresholding operator with threshold $\beta_1\sigma$. In the second iteration, we use a Wiener filter where the variances of the signal component are estimated using the transformed coefficients of the group of patches from the first iterate (for more details see for example [4]).

D. Discussion

Table I summarizes the main differences between the models described. Video NL-Bayes and VNLDCT assume that patches in the group are IID samples of a Gaussian model. They work by applying Bayesian shrinkage operators in a transformed domain (the principal directions of the covariance matrix). VNLDCT assumes a fixed basis of principal directions (the DCT), and VNLB seeks to estimate an optimal basis (the PCA basis). BM4D-OF uses a fixed basis, but uses a model for the whole group of patches. As such, it takes into account correlations between patches in the group.

An additional difference is in how the mean patch of the group is handled. For VNLB and VNLDCT the mean patch μ is a parameter of the Gaussian model. It is estimated as the sample average of the noisy patches. For BM4D-OF the mean patch is just the DC component of the Haar transform (the transform in the fourth dimension). Since it is part of the estimated signal it is filtered by the shrinkage operators.

The asymptotic complexity of the group estimation is $\mathcal{O}(nd)$ for VNLDCT and VBM4D-OF, and $\mathcal{O}(nd^2 + rd^2 + nrd)$ for VNLB (where r is a maximal rank parameter for the algorithm). For VNLB the most expensive operations are the computation of the covariance matrix and of its r principal directions. This methods scales badly with the patch size due to the quadratic dependence on the patch dimensionality.

III. Results

The three methods have the same parameters. For iteration i ($i = 1, 2$) of the method we need to specify: the spatial and temporal sizes of the patch and of the search region; the number of similar patches n_i; the maximum distance threshold τ_i and the filtering strength coefficient β_i.

We fix the size of the patch at $10 \times 10 \times 2$ and the size of the search region at $21 \times 21 \times 9$. We also fix $\beta_2 = 1$. This leaves us with five parameters to tune. For video NL-Bayes, we use the optimal parameters reported in [7] for the VNLB-H variant of the method and the patch of size $10 \times 10 \times 2$. For VNLDCT and BM4D-OF, we do a random search on the 5D parameter space $n_1, n_2, \tau_1, \tau_2, \beta_1$ and select the ones that maximize the average MSE over four 20 frame sequences.

We tested our methods on 7 grayscale videos of resolution of 960×540 with 100 frames, obtained from the DERF Video database https://media.xiph.org/video/derf/. The original videos are RGB of size 1920×1080. We converted them to grayscale (channel average), then down-sampled them by a factor two (after the application of an anti-aliasing filter). The sequences and results for the different methods are available at: `https://goo.gl/Wng5JW`

The results obtained for the three methods are shown Table II, together with the results obtained with V-BM3D [15], V-BM4D [3] and SPTWO [8]. Both V-BM3D and V-BM4D are extensions of BM3D to video, the first stacks 2D patches extracted from the video, while the second creates stacks of 3D spatio-temporal patches which are motion compensated (as opposed to BM4D-OF which uses rectangular 3D patches)[1]. SPTWO is an extension of NL-Bayes to video. To denoise a frame, it registers the neighboring frames to it (this requires computing the optical flow from the target frame to all its neighboring frames, typically around 10 frames). The method then works with 3D patches extracted from this motion compensated volume.

The best performance is attained by VNLB, although the gap is considerably smaller for higher noise levels. BM4D-OF shows overall a very good performance. It outperforms V-BM3D and V-BM4D and has an average performance comparable to SPTWO. The compared methods outperform the ones in the literature for noise 20 and 40. In particular,

[1]The similar method BM4D [4] uses rectangular patches as BM4D-OF. We do not include it in the comparison since it was shown in [4] that V-BM4D performs better in video.

TABLE II

QUANTITATIVE DENOISING RESULTS FOR SEVEN GRAYSCALE TEST SEQUENCES OF SIZE 960×540. WE SHOW PSNR AND SSIM.

σ	Method	crowd	park joy	pedestrians	station	sunflower	touchdown	tractor	average
10	SPTWO	36.57 / .9651	35.87 / .9570	41.02 / .9725	41.24 / .9697	42.84 / .9824	40.45 / .9557	38.92 / .9701	39.56 / .9675
	V-BM3D-np	35.76 / .9589	35.00 / .9469	40.90 / .9674	39.14 / .9651	40.13 / .9770	39.25 / .9466	37.51 / .9575	38.24 / .9599
	V-BM4D-mp	36.05 / .9535	35.31 / .9354	40.61 / .9712	40.85 / .9466	41.88 / .9696	39.79 / .9440	37.73 / .9533	38.88 / .9534
	VNLB	**37.24 / .9702**	**36.48 / .9622**	**42.23 / .9782**	**42.14 / .9771**	**43.70 / .9850**	**41.23 / .9615**	**40.20 / .9773**	**40.57 / .9731**
	VNLDCT	35.96 / .9596	35.23 / .9479	41.37 / .9744	41.12 / .9673	42.43 / .9814	40.37 / .9552	39.00 / .9706	39.35 / .9652
	BM4D-OF	35.86 / .9616	35.15 / .9490	41.55 / .9757	41.75 / .9711	42.76 / .9827	40.62 / .9583	39.09 / .9715	39.54 / .9671
20	SPTWO	32.94 / .9319	32.35 / **.9161**	37.01 / .9391	38.09 / .9461	38.83 / .9593	**37.55 / .9287**	35.15 / .9363	35.99 / .9368
	V-BM3D-np	32.34 / .9093	31.50 / .8731	37.06 / .9423	35.91 / .9007	36.25 / .9393	36.17 / .9065	33.53 / .8991	34.68 / .9100
	V-BM4D-mp	32.40 / .9126	31.60 / .8832	36.72 / .9344	36.84 / .9224	37.78 / .9517	36.44 / .9034	33.95 / .9104	35.10 / .9169
	VNLB	**33.49 / .9335**	**32.80** / .9154	**38.61 / .9583**	**38.78 / .9470**	**39.82 / .9698**	37.47 / .9220	**36.67 / .9536**	**36.81 / .9428**
	VNLDCT	32.62 / .9218	31.94 / .8992	37.88 / .9519	37.88 / .9383	38.92 / .9648	37.15 / .9204	35.58 / .9413	36.00 / .9340
	BM4D-OF	32.52 / .9253	31.79 / .8987	38.14 / .9558	38.22 / .9408	39.38 / .9688	37.51 / .9258	35.76 / .9442	36.19 / .9371
40	SPTWO	29.02 / .8095	28.79 / .8022	31.32 / .7705	32.37 / .7922	32.61 / .7974	31.80 / .7364	30.61 / .8223	30.93 / .7901
	V-BM3D-np	28.73 / .8295	27.93 / .7663	33.00 / .8828	32.57 / .8239	32.39 / .8831	33.38 / .8624	29.80 / .8039	31.11 / .8360
	V-BM4D-mp	28.72 / .8339	27.99 / .7751	32.62 / .8683	32.93 / .8441	33.66 / .8999	33.68 / .8603	30.20 / .8205	31.40 / .8432
	VNLB	**29.88 / .8682**	**29.28 / .8309**	**34.68** / .9167	**34.65 / .8871**	35.44 / .9329	34.18 / .8712	**32.58 / .8921**	**32.95 / .8856**
	VNLDCT	29.32 / .8511	28.73 / .8103	34.14 / .9013	34.23 / .8745	34.92 / .9198	33.76 / .8479	31.95 / .8770	32.43 / .8688
	BM4D-OF	29.39 / .8628	28.70 / .8156	34.57 / **.9184**	34.46 / .8819	**35.78 / .9399**	**34.67 / .8858**	32.09 / .8808	32.81 / .8836

the difference in performance between BM4D-OF and both V-BM3D and V-BM4D is remarkable. The main reason is the use of a large patch ($10 \times 10 \times 2$) which does not require motion compensation. V-BM3D uses $8 \times 8 \times 1$, and V-BM4D uses patch trajectories of size $8 \times 8 \times 9$. Although these patches have a higher dimensionality, they are constructed by matching blocks of $8 \times 8 \times 1$. We have tested if better results can be obtained by further increasing the patch size of BM4D-OF, but we found almost no gain.

VNLDCT has a PSNR slightly lower than BM4D-OF, and its results are of inferior visual quality. For example, for the sequence shown Fig. 1 VNLDCT has introduced ringing artifacts. The fact that VNLDCT does not filter the mean patch forces the method to use a larger number of similar patches n_1, n_2 (we use n_2 is 16 for BM4D-OF and 70 for VNLDCT). This causes some loss of detail in the output. The running times for VNLB are approximately 140s per frame (of size 960×540), for BM4D-OF are around 36s and for VNLDCT of 50s per frame [2].

Fig. 1. Detail of the results obtained with BM4D-OF, VNLDCT and VNLB on the *pedestrian* sequence for a noise of standard deviation $\sigma = 40$. The contrast has been linearly enhanced to make the differences between the methods more visible.

IV. CONCLUSION

In this work we compared three models for patch-based video denoising in equal conditions (same strategy for patch search, same patch sizes, and same codebase). We found that using optimal adaptive transforms such as VNLB indeed achieves better results. But this gain in denoising quality reduces with higher noise levels, and comes at the price of a factor of four in the running time. Meanwhile, BM4D-OF method yields very competitive results with a much lower computing cost. Future research will be focused on exploring the theoretical counterpart of the observed trade-offs from the viewpoint of estimation theory.

[2] The three methods are implemented in C++ from the same codebase. Times computed on an Intel(R) Xeon(R) CPU at 2.60GHz.

REFERENCES

[1] K. Dabov, A. Foi, V. Katkovnik, and K. O. Egiazarian, "Image denoising with block-matching and 3D filtering," in *Proc. SPIE 6064, Electronic Imaging 2006*, vol. 6064, 2006. [Online]. Available: http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.643267

[2] K. Dabov, A. Foi, and K. O. Egiazarian, "Video denoising by sparse 3D transform-domain collaborative filtering," in *EUSIPCO*, 2007, pp. 145–149.

[3] M. Maggioni, G. Boracchi, A. Foi, and K. O. Egiazarian, "Video denoising, deblocking, and enhancement through separable 4-D nonlocal spatiotemporal transforms," *IEEE Transactions on Image Processing*, vol. 21, no. 9, pp. 3952–3966, 2012.

[4] M. Maggioni, V. Katkovnok, K. O. Egiazarian, and A. Foi, "A nonlocal transform-domain filter for volumetric data denoising and reconstruction," *IEEE Transactions on Image Processing*, vol. 22, no. 1, pp. 119–133, 2013.

[5] W. Li, J. Zhang, and Q.-H. Dai, "Video denoising using shape-adaptive sparse representation over similar spatio-temporal patches," *Signal Processing: Image Communication*, vol. 26, pp. 250–265, April 2011.

[6] M. Lebrun, A. Buades, and J.-M. Morel, "A Nonlocal Bayesian Image Denoising Algorithm," *SIAM Journal on Imaging Sciences*, vol. 6, no. 3, pp. 1665–1688, 2013.

[7] P. Arias and J.-M. Morel, "Video denoising via empirical Bayesian estimation of space-time patches," *Journal of Mathematical Imaging and Vision*, 2017.

[8] A. Buades, J.-L. Lisani, and M. Miladinović, "Patch-based video denoising with optical flow estimation," *IEEE Transactions on Image Processing*, vol. 25, no. 6, pp. 2573–2586, June 2016.

[9] P. Arias and J.-M. Morel, "Towards a bayesian video denoising method," in *Advanced Concepts for Intelligent Vision Systems*, 2015. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-25903-1_10

[10] T. Ehret, P. Arias, and J. Morel, "Global patch search boosts video denoising," in *Proc. of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP).*, 2017, pp. 124–134. [Online]. Available: https://doi.org/10.5220/0006175601240134

[11] G. Yu, G. Sapiro, and S. Mallat, "Solving inverse problems with piecewise linear estimators: From gaussian mixture models to structured sparsity," *Image Processing, IEEE Transactions on*, vol. 21, no. 5, pp. 2481–2499, May 2012.

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

[13] J. Sánchez Pérez, E. Meinhardt-Llopis, and G. Facciolo, "TV-L1 Optical Flow Estimation," *Image Processing On Line*, vol. 3, no. 1, pp. 137–150, jul 2013. [Online]. Available: http://www.ipol.im/pub/art/2013/26/http://www.ipol.im/pub/art/2013/26/?utm{_}source=doihttp://dx.doi.org/10.5201/ipol.2013.26

[14] G. Yu and G. Sapiro, "DCT image denoising: a simple and effective image denoising algorithm," *Image Processing On Line*, vol. 1, oct 2011. [Online]. Available: http://www.ipol.im/pub/art/2011/ys-dct/?utm{_}source=doi

[15] K. Dabov, A. Foi, V. Katkovnik, and K. O. Egiazarian, "Color image denoising via sparse 3d collaborative filtering with grouping constraint in luminance-chrominance space," in *Proc. IEEE International Conference on Image Processing*, 2007.

[16] D. L. Donoho and J. M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, no. 3, pp. 425–455, 1994. [Online]. Available: http://biomet.oxfordjournals.org/content/81/3/425.abstract

[17] R. Oktem, L. P. Yaroslavsky, and K. O. Egiazarian, "Signal and image denoising in transform domain and wavelet shrinkage: A comparative study," in *9th European Signal Processing Conference (EUSIPCO 1998)*, sep 1998, pp. 1–4.

[18] L. P. Yaroslavsky, K. O. Egiazarian, and J. T. Astola, "Transform domain image restoration methods: review, comparison, and interpretation," in *Proc. SPIE 4304, Electronic Imaging 2001*, E. R. Dougherty and J. T. Astola, Eds., may, pp. 155–169.