NON-LOCAL KALMAN: A RECURSIVE VIDEO DENOISING ALGORITHM

*Thibaud Ehret, Jean-Michel Morel, Pablo Arias**

CMLA, ENS Cachan, CNRS, Université Paris-Saclay, 94235 Cachan, France

ABSTRACT

In this article we propose a new recursive video denoising method with high performance. The method is recursive and uses only the current frame and the previous denoised one. It considers the video as a set of overlapping temporal patch trajectories. Following a Bayesian approach each trajectory is modeled as linear dynamic Gaussian model and denoised by a Kalman filter. To estimate its parameters, similar patches are grouped and their trajectories are considered as sharing the same model parameters. The filtering is mainly temporal; non-local spatial similarity is only used to estimate the parameters. This temporally causal method obtains results comparable (in terms of PSNR and SSIM) to state-of-the-art methods using several frames per frame denoised, but with a higher temporal consistency.

***Index Terms*—** Recursive filtering, Video denoising, Patch-based methods

1. INTRODUCTION

There are two current trends in video denoising. The first one aims at producing the best video denoising possible whatever the computation required, while the second trend focuses on producing the fastest causal video denoising algorithm with the goal of real-time processing.

In terms of output quality the state of the art is achieved by patch-based methods [1, 2, 3, 4, 5, 6]. They exploit the self-similarity of natural images and videos, namely that most patches have several similar patches around them (spatially and temporally). Each patch is denoised using its similar patches, which are searched for in a region around it. The search region generally is a spatio-temporal cube, but more sophisticated search strategies have also been used. Because of the use of such neighborhoods these methods are called *non-local*. While these algorithms perform very well, they often are unpractical: they use a frame's past and future and therefore can only be used off-line. Because of their complexity they are unfit for high resolution video processing.

Fast algorithms rely on much simpler principles which can be implemented efficiently on GPUs or FPGAs. For example [7] relies on a bilateral filter and a Kalman filter to produce a real-time video denoising algorithm. A recursive version of the non-local means algorithm is proposed in [8]. These methods use simple denoising strategies yet generally result in poor denoising quality for high noise levels.

Convolutional networks have been successfully applied to image denoising, and also to other video processing tasks such as deblurring [9] or video synthesis. Their application to video denoising has been limited so far. In [10] a recurrent architecture is proposed, but the results are quite below the state of the art. Recently [11] focused on the related problem of image burst denoising reporting very good results.

In this work we present a video denoising method that tries to close the gap between high-quality but slow methods and those efficient but of lower quality. It is much faster than most state-of-the art algorithms (yet not real-time) but in contrast to them it is causal temporally and recursive, i.e. it uses only the current frame and the previous denoised frame. In spite of these strong design constraints it achieves a state-of-the-art performance in image quality.

2. PROPOSED METHOD

In the following we denote by u a clean video which has been contaminated by an additive Gaussian white noise n of (known) standard deviation σ, i.e. only $v = u+n$ is observed. This white noise model is sufficient to deal with realistic Poisson noise. Indeed, this noise can be reduced to nearly white Gaussian by a variance stabilizing transform [12]. The video u, respectively v, is made of f frames indexed between 0 and $f-1$; the frames are denoted by u_t, respectively v_t, with $t \in [\![0, f-1]\!]$. We will work with two dimensional square patches of size $s \times s$, a patch will be denoted by a bold lower case letter such as $\boldsymbol{p}$, considered as a vector of dimension s^2.

2.1. Framework

Our method works by temporal filtering along patch trajectories using a Kalman filter associated to each trajectory. The Kalman filter requires for its operation to keep track of the state covariance matrix. The state in our case is the clean patch that we want to estimate (arranged as a vector of s^2 components), and the state covariance is a $s^2 \times s^2$ matrix. We use a simple dynamic model for the temporal evolution of the patch trajectories, which requires the estimation of a single parameter at each time step: the *state transition covariance matrix*. It models how a (clean) patch can vary from

*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.

one frame to the next. Estimating the state transition covariance matrix is in general a difficult problem. We resort to non-locality for obtaining a reliable estimate. To that aim, we group trajectories based on the similarity of the first patch, and assume that these trajectories share the same model. The covariance matrix is obtained from the statistics of the trajectories in the group. This assumption also allows us to reduce the memory consumption of the method, since all the trajectories in the group, because they share the same transition covariance, also share the state covariance. Because there is still a small high frequency residual noise after the NL-Kalman pass, the multiscale DCT denoising algorithm [13] is applied to each frame as a post-processing.

The definition of patch trajectories, and their computation from an optical flow is explained in Section 2.2. Patch trajectories can be terminated if an occlusion happens (or more generally due to a registration error). This is explained in section Section 2.5. The spatial denoising algorithm is responsible for initializing the groups of trajectories, and is explained in Section 2.3. Finally, the key of our contribution, namely the non-local Kalman filters running on groups of trajectories, is explained in Section 2.4.

2.2. Patch trajectories

A patch trajectory is temporal sequence of $s \times s$ patches $\boldsymbol{q}_0, \boldsymbol{q}_1, \ldots, \boldsymbol{q}_t$ extracted from consecutive frames in the video. The centers of these patches follow a motion trajectory, which is estimated using an optical flow algorithm.

Suppose we know the trajectory of a patch until frame $t-1$, $\boldsymbol{q}_0, \boldsymbol{q}_1, \ldots, \boldsymbol{q}_{t-1}$. The next patch in the trajectory at frame t is computed by tracing the optical flow between frames t and $t+1$. We assume that all the pixels in the patch move with the same optical flow as the patch center. This is a crude approximation but it allows for a simpler dynamic model.

We use an optical flow to compute the patch trajectories because it is dense and is not restricted to any particular type of motion (as opposed to e.g. global parametric models). In particular, we used the implementation of [14] of the TV-L1 optical flow [15]. Any other motion model can be used, as long as it provides a dense set of correspondences. Optical flow is less robust to noise than global parametric models. To gain robustness to noise (and speed), we compute it on a downscaled version of the frames, similar to [16].

2.3. Spatial denoising: creation of groups of trajectories

The spatial denoising is used when no temporal information from the previous frame is available (e.g. in a dis-occluded area). The goal here is not only to denoise these areas, but initialize the groups of trajectories for the temporal denoising of these areas in the future frames. We use a single step of the NL-Bayes denoising algorithm [17], which has the benefit of computing groups similar patches and their covariance matrices (needed for the Kalman filters) as part of the denoising.

Fig. 1. Evolution of the quality of the denoising during the warm-up stage. Artifacts due to the spatial denoising quickly disappear after few iterations of the temporal filtering.

Given a patch $\boldsymbol{q}$ of the noisy frame v_t, and the corresponding unknown patch $\boldsymbol{p}$ of the clean frame u_t, the following Gaussian linear model of $\boldsymbol{p}$ and $\boldsymbol{q}$ is assumed: $\mathbb{P}(\boldsymbol{p}) = \mathcal{N}(\boldsymbol{\mu}, C)$ and $\mathbb{P}(\boldsymbol{q}|\boldsymbol{p}) = \mathcal{N}(\boldsymbol{p}, \sigma I)$. Once the mean patch $\boldsymbol{\mu}$ and the covariance matrix C have been estimated from the noisy video, the MAP estimate $\hat{\boldsymbol{p}}$ given a noisy patch $\boldsymbol{q}$ is obtained as in [17]: $\hat{\boldsymbol{p}} = \boldsymbol{\mu} + C(C + \sigma^2 I)^{-1}(\boldsymbol{q} - \boldsymbol{\mu})$.

The parameters of the *a priori* model are learned from the noisy frame. For each noisy patch $\boldsymbol{q}$, N similar patches are selected from the frame. Let $\boldsymbol{q}_i$, $i = 1, \ldots, N$ be the set of patches similar to $\boldsymbol{q}$ (with $\boldsymbol{q}_1 = \boldsymbol{q}$). The estimates for $\boldsymbol{\mu}$ and C are given by $\hat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{q}_i$ and $\hat{C} = \frac{1}{N}\sum_{i=1}^{N} \overline{\boldsymbol{q}}_i \overline{\boldsymbol{q}}_i^T - \sigma^2 I$, where $\overline{\boldsymbol{q}}_i = \boldsymbol{q}_i - \hat{\boldsymbol{\mu}}$ are the centered patches.

2.4. Temporal filtering

For denoising frame v_t, we receive from the previous frame a set of groups of trajectories. The union of all trajectories in the groups covers the previous frame. To denoise the frame t, we loop on the groups updating the parameters of the Kalman filter and operating it to estimate the clean patches of the trajectories in the group at frame t.

Let $G = \{\hat{\boldsymbol{p}}_{t-1,1}, \ldots, \hat{\boldsymbol{p}}_{t-1,N}\}$ be a group of trajectories. For each patch $\hat{\boldsymbol{p}}_{t-1,i}$ we have a noisy observation $\boldsymbol{q}_{t,i}$. The registration test (explained in the next section) compares $\hat{\boldsymbol{p}}_{t-1,i}$ and $\boldsymbol{q}_{t,i}$ and determines if the registration is correct or not. The trajectories that do not pass this test are terminated and removed from the group. In the following we assume that the N trajectories in the group passed the test. For such patches, we have a prediction, namely the previous denoised patch in the trajectory.

We assume that the evolution of the patch along a trajectory follows a simple linear dynamic Gaussian model, described as follows:

$$\boldsymbol{p}_{t+1,i} = \boldsymbol{p}_{t,i} + \boldsymbol{w}_{t,i} \text{ with } \boldsymbol{w}_{t,i} \sim \mathcal{N}(\boldsymbol{0}, C_t) \quad (1)$$

$$\boldsymbol{q}_{t,i} = \boldsymbol{p}_{t,i} + \boldsymbol{n}_{t,i} \text{ with } \boldsymbol{n}_{t,i} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 I). \quad (2)$$

Here $\boldsymbol{w}_{t,i} \sim \mathcal{N}(0, C_t)$ is the process noise, modelling the variations of a patch from one frame to the next. The only model parameters that need to be estimated are the state transition covariances C_t associated to the group. It is easy to verify that $\mathbb{E}\{(\boldsymbol{q}_{t,i} - \boldsymbol{q}_{t-1,i})(\boldsymbol{q}_{t,i} - \boldsymbol{q}_{t-1,i})^T\} = C_t + 2\sigma^2 I$.

Table 1. Quantitative comparison with VBM3D and VBM4D on grayscale sequences. NL-Kalman (oracle) corresponds to the proposed method using an optical flow computed on the clean sequence. Results with a star were computed using the binary provided by the author. Best results are in bold (results from the oracle are excluded). VBM3D has the lowest running times, followed by NL-Kalman and finally VBM4D. Other state-of-the-art methods like SPTWO [5] and video NL-Bayes [3] can achieve better results, but with a significantly higher running time. See text for details.

σ	Method	Bus	Foreman	Pedestrian_area	Crowd_run	Touchdown_pass	Station2	Average
10	VBM3D*	33.32/.7824	**37.40**/.6681	**40.78**/.6577	35.62/.8017	39.08/.6103	38.92/.7266	37.52/.7078
	VBM4D-np*	**33.39**/.8237	37.39/**.6871**	40.56/**.7463**	**35.69**/**.8457**	**39.60**/.6752	**39.93**/.7746	**37.76**/.7588
	NL-Kalman	33.34/**.8502**	36.16/.6782	38.67/.7420	34.29/.8383	38.82/**.6940**	39.91/**.7916**	36.86/**.7657**
	NL-Kalman (oracle)	*33.87/.8713*	*36.93/.7230*	*39.23/.7592*	*34.64/.8514*	*39.58/.7433*	*40.50/.8059*	*37.46/.7923*
20	VBM3D*	29.57/.6064	34.60/.5763	**36.93**/.5579	**32.22**/.7122	36.09/.4703	35.45/.5689	34.14/.5820
	VBM4D-np*	29.55/.6856	**34.61**/**.6073**	36.75/**.6468**	32.07/.7439	**36.41**/.4795	36.23/.6395	**34.27**/.6338
	NL-Kalman	**29.58**/**.7291**	33.19/.5844	35.61/.6444	30.89/**.7478**	35.91/**.5181**	**36.81**/**.6868**	33.66/**.6518**
	NL-Kalman (oracle)	*30.43/.7752*	*34.18/.6301*	*36.45/.6738*	*31.44/.7746*	*36.99/.6135*	*37.46/.7116*	*34.49/.6965*
30	VBM3D*	**27.59**/.4995	32.77/.5224	34.44/.4869	**30.14**/.6394	34.55/.3906	33.36/.4536	32.14/.4987
	VBM4D-np*	27.53/.5988	**32.91**/**.5612**	**34.45**/**.5745**	29.95/.6704	**34.76**/.3801	34.14/.5420	**32.29**/.5545
	NL-Kalman	27.30/**.6327**	31.27/.5335	33.27/.5680	28.64/**.6708**	33.91/**.4034**	**34.73**/**.5986**	31.52/**.5678**
	NL-Kalman (oracle)	*28.48/.6993*	*32.50/.5802*	*34.43/.6102*	*29.44/.7078*	*35.20/.5186*	*35.46/.6338*	*32.59/.6250*

By assuming that the patches in a group are independent realizations of the same dynamic model, we can estimate the expectation as the sample covariance matrix of the *innovation* vectors $\boldsymbol{q}_{t,i} - \boldsymbol{q}_{t-1,i}$ of the trajectories in the group. We thus estimate C_t as the following spatio-temporal average:

$$C_t = \beta C_{t-1} + \frac{(1-\beta)}{2} \cdot \cdot \left(\sum_{i=1}^{N} \frac{(\boldsymbol{q}_{t,i} - \boldsymbol{q}_{t-1,i})(\boldsymbol{q}_{t,i} - \boldsymbol{q}_{t-1,i})^T}{N-1} - 2\sigma^2 I \right),$$

To increase the temporal stability of the estimate, we combine the non-local average over the group with the previous estimate C_{t-1}, with a forgetting factor $\beta \in [0,1]$. The initial C_0 is set as the covariance of the spatial denoising. We estimate the patches and update their covariance with the Kalman filter equations [18]:

$$\text{Predicted estimate cov. } P'_t = P_{t-1} + C_t \quad (3)$$

$$\text{Optimal Kalman gain } K_t = P'_t(P'_t + \sigma^2 I)^{-1} \quad (4)$$

$$\text{Update state } \hat{\boldsymbol{p}}_{t,i} = \hat{\boldsymbol{p}}_{t-1,i} + K_t(\boldsymbol{q}_t - \hat{\boldsymbol{p}}_{t-1,i}) \quad (5)$$

$$\text{Updated estimate cov. } P_t = P'_t - K_t(P'_t + \sigma^2 I)K_t^T \quad (6)$$

Here P_t is the state covariance matrix, it needs to be stored with the group of patches (as the state transition covariance C_t). The dynamic model is initialized using the result of the spatial denoising on the patch group. In particular the initial state covariance matrix is taken as the covariance computed by the NL-Bayes spatial denoising algorithm.

The proposed model can be interpreted as a dynamical version of the video NL-Bayes denoising method [3]. Their similar 3D patches (of size around $7 \times 7 \times 2$) are assumed to be normally distributed. The parameters of these Gaussian distributions are estimated from the set of similar patches, gathered from past and future frames. The estimated covariance matrix represents the modes of variation of the population from the estimated mean patch. Instead our Gaussian dynamical models represent temporal variations along patch trajectories.

2.5. Registration quality assessment

We now address the registration quality assessment, to detect occlusions and errors in the optical flow. The quality must be assessed for each patch trajectory individually. The idea is to compare the denoised patch (the prediction in the Kalman filter) to the new noisy observation. Let $\boldsymbol{q}_t$ be the patch candidate and $\boldsymbol{p}_{t-1}$ the result of the denoising for the corresponding patch at the previous frame. The assumption underlying the registration quality assessment is that $\boldsymbol{q}_t$ is a noisy version of $\boldsymbol{p}_{t-1}$ (Eq. (2)). Therefore $\frac{\|\boldsymbol{p}-\boldsymbol{q}\|_2^2}{\sigma^2} \sim \chi^2(d)$. Our goal is to reject patches that are too different, as the dynamic system can only deal with small changes. The optimal test to assess the quality of the matching corresponds to $\frac{\|\boldsymbol{p}-\boldsymbol{q}\|_2^2}{\sigma^2} \geq c$.

In order to choose the best possible c, one can use a statistical approach like in [19, 20, 21, 22, 23, 24] based on a number of false alarm (NFA). We would like the patch $\boldsymbol{p}$ to be rejected if and only if we are confident enough that it is not a good match and not because of the noise *i.e.* $N\mathbb{P}\left(\frac{\|\boldsymbol{p}-\boldsymbol{q}\|_2^2}{\sigma^2} \geq c\right) \leq \varepsilon$ where ε corresponds to the number of false matches that can be accepted and N to the number of time a patch is added to trajectory. This leads to $c \geq G^{-1}\left(1 - \frac{\varepsilon}{N}\right)$ where G^{-1} the inverse cdf of a chi square distribution of dimension d. A sound value for N is the number of pixels of a frame. Indeed, fixing $\epsilon = 1$ amounts then to admit at most one false matching on average per frame.

Fig. 2. Denoising results when the sequence is corrupted by noise of standard deviation 30. From left to right: Denoising using NL-Kalman, NL-Kalman with the oracle optical flow, VBM3D, VBM4D and the original. From top to bottom: Crop from the sequence `crowd_run` and `pedestrian_area`. Overall the proposed method preserves more details as can be seen both in the tree and in the text. Results best viewed zoomed.

3. EXPERIMENTS

We compared the performance of our algorithm against the state-of-the-art denoising algorithms VBM3D and VBM4D. Our causal recursive method requires a couple of frames to warm up. Therefore we remove the first five frames in the computation of the PSNR and SSIM to get a fair comparison. Figure 1 shows the quick improvement of the denoising quality in the first three frames used as warm up. Table 1 presents denoising results on 1920×1080 sequences from Derf's video database[1] that have been converted to gray by averaging the RGB channels and downscaled by two so as to ensure that they contain little noise. A Gaussian blur was applied before downscaling to avoid aliasing.

Overall the proposed method performs better than VBM3D and VBM4D in SSIM but shows a lower PSNR. The main limitation of NL-Kalman stems from the optical flow. Estimating the optical flow from noisy data is challenging. This causes miss-registration errors which explains the observed drop in PSNR performance. To evaluate this we compare the denoising results obtained with optical flow computed on the noisy data to those obtained from ground truth optical flow. In spite of this, the result of NL-Kalman shows a much higher visual quality, in terms of temporal consistency and in the amount of recovered details. This explains why the difference between the SSIM measure, where NL-Kalman leads, and PSNR, where the average difference is about 0.7db in favor of VBM4D. A comparable result in terms of temporal consistency is attained with SPTWO [5], a more complex method using 10 frames for each frame denoised. SPTWO is also based on the optical flow, but is not recursive. To denoise each frame it computes the optical flow between the target frame and n neighboring frames ($n/2$ past plus $n/2$ future frames). We computed SPTWO with $n = 8$ for $\sigma = 30$ and obtained an average PSNR of 32.45 (SSIM .6114).

Among the algorithms compared, the faster one is VBM3D. Our current implementation in non-optimized C++ code is between two to three times slower than VBM3D. The main reason for this is that the number of groups grows as time evolves, since new groups are created for covering patches in dis-occluded areas. We are currently working on faster versions of the algorithm by limiting the number of groups.

4. CONCLUSION AND FUTURE WORK

We presented a novel patch-based video denoising method and showed that it is competitive with the state of the art. It is based on temporal filtering along trajectories of patches. The temporal filtering is carried out by Kalman filters whose parameters are estimated non-locally. The resulting method is recursive, as to denoise one frame it uses solely information from the previous frame. Results show a performance comparable to more complex patch-based methods that use around ten frames to denoise each single frame. The main limitation of the method is that it relies heavily on the optical flow. Slight errors in registration cause a drop in PSNR, even if visual quality of the result is much superior to the one of related algorithms, both in terms of details and temporal consistency. This work also shows the limits of the PSNR and SSIM as quantitative measures for video quality.

[1] https://media.xiph.org/video/derf/

5. REFERENCES

[1] Kostadin Dabov, Alessandro Foi, and Karen Egiazarian, "Video denoising by sparse 3D transform-domain collaborative filtering," in *EUSIPCO*, 2007.

[2] Matteo Maggioni, Giacomo Boracchi, Alessandro Foi, and Karen Egiazarian, "Video denoising, deblocking, and enhancement through separable 4-D nonlocal spatiotemporal transforms," *IEEE Transactions on Image Processing*, 2012.

[3] Pablo Arias and Jean-Michel Morel, "Video denoising via empirical bayesian estimation of space-time patches," *Journal of Mathematical Imaging and Vision*, vol. 60, no. 1, pp. 70–93, Jan 2018.

[4] Thibaud Ehret, Pablo Arias, and Jean-Michel Morel, "Global patch search boosts video denoising," in *International Conference on Computer Vision Theory and Applications*, 2017.

[5] Antoni Buades, Jose-Luis Lisani, and Marko Miladinović, "Patch-based video denoising with optical flow estimation," *IEEE Transactions on Image Processing*, vol. 25, no. 6, pp. 2573–2586, June 2016.

[6] Bihan Wen, Yanjun Li, Luke Pfister, and Yoram Bresler, "Joint adaptive sparsity and low-rankness on the fly: an online tensor reconstruction scheme for video denoising," in *IEEE ICCV*, 2017.

[7] Sergio G. Pfleger, Patricia D. M. Plentz, Rodrigo C. O. Rocha, Alyson D. Pereira, and Márcio Castro, "Real-time video denoising on multicores and gpus with kalman-based and bilateral filters fusion," *Journal of Real-Time Image Processing*, Feb 2017.

[8] Redha A Ali and Russell C Hardie, "Recursive non-local means filter for video denoising," *EURASIP Journal on Image and Video Processing*, 2017.

[9] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang, "Deep video deblurring for hand-held cameras," in *IEEE CVPR*, 2017.

[10] Xinyuan Chen, Li Song, and Xiaokang Yang, "Deep rnns for video denoising," in *Applications of Digital Image Processing*, 2016.

[11] Clément Godard, Kevin Matzen, and Matt Uyttendaele, "Deep burst denoising," *arXiv*, 2017.

[12] Bo Zhang, MJ Fadili, J-L Starck, and J-C Olivo-Marin, "Multiscale variance-stabilizing transform for mixed-poisson-gaussian processes and its applications in bioimaging," in *IEE ICIP 2007*, 2007.

[13] Nicola Pierazzo, Jean-Michel Morel, and Gabriele Facciolo, "Multi-Scale DCT Denoising," *Image Processing On Line*, 2017.

[14] Javier Snchez Prez, Enric Meinhardt-Llopis, and Gabriele Facciolo, "TV-L1 Optical Flow Estimation," *Image Processing On Line*, 2013.

[15] Christopher Zach, Thomas Pock, and Horst Bischof, "A duality based approach for realtime tv-l 1 optical flow," in *Joint Pattern Recognition Symposium*. Springer, 2007.

[16] Ziwei Liu, Lu Yuan, Xiaoou Tang, Matt Uyttendaele, and Jian Sun, "Fast burst images denoising," *ACM Transactions on Graphics (TOG)*, 2014.

[17] Marc Lebrun, Antoni Buades, and Jean-Michel Morel, "A nonlocal bayesian image denoising algorithm," *SIAM Journal on Imaging Sciences*, 2013.

[18] Rudolph Emil Kalman, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, 1960.

[19] Agnes Desolneux, Lionel Moisan, and Jean-Michel Morel, *From gestalt theory to image analysis: a probabilistic approach*, Springer Science & Business Media, 2007.

[20] Rafael Grompone Von Gioi, Jeremie Jakubowicz, Jean-Michel Morel, and Gregory Randall, "Lsd: A fast line segment detector with a false detection control," *IEEE Trans. PAMI*, 2010.

[21] V Patraucean, P Gurdjos, and R Grompone von Gioi, "A parameterless ellipse and line segment detector with enhanced ellipse fitting," in *ECCV*, 2012.

[22] José Lezama, Rafael Grompone von Gioi, Gregory Randall, and Jean-Michel Morel, "Finding vanishing points via point alignments in image primal and dual domains," in *CVPR*, 2014.

[23] Lionel Moisan and Bérenger Stival, "A probabilistic criterion to detect rigid point matches between two images and estimate the fundamental matrix," *Int. J. Comput. Vis.*, 2004.

[24] Viorica Patraucean, Rafael Grompone von Gioi, and Maks Ovsjanikov, "Detection of mirror-symmetric image patches," in *CVPR*, 2013.