

PRNU PATTERN ALIGNMENT FOR IMAGES AND VIDEOS BASED ON SCENE CONTENT

F. Bellavia, M. Iuliani, M. Fanfani, C. Colombo, A. Piva

Università degli Studi di Firenze, Via di S. Marta 3, 50139, Firenze, Italy

ABSTRACT

This paper proposes a novel approach for registering the PRNU pattern between different camera acquisition modes by relying on the imaged scene content. First, images are aligned by establishing correspondences between local descriptors: The result can then optionally be refined by maximizing the PRNU correlation. Comparative evaluations show that this approach outperforms those based on brute-force and particle swarm optimization in terms of reliability, accuracy and speed. The proposed scene-based approach for PRNU pattern alignment is suitable for video source identification in multimedia forensics applications.

Index Terms— PRNU, SIFT matching, image registration, video stabilization

1. INTRODUCTION

The Photo Response Non-Uniformity (PRNU) [1] of digital sensors is a special fixed-pattern noise, exploited in image forensics as fingerprint to develop robust methods for source attribution [2]. PRNU fingerprints are extracted pixelwise, and are hence best generated and compared at native camera resolution [3]. Due to their high sensitivity to any pixel misalignment, PRNU fingerprints are particularly difficult to compare when dealing with stabilized videos, that are typically obtained from a scaled, translated and/or rotated portion of the full sensor area [4]. Current approaches to the problem attempt to find accurate estimates of the underlying geometrical transformation by maximizing the PRNU correlation either by brute-force search [5–7] or, more recently, by particle swarm optimization [8]. Nevertheless, such kind of approaches can be computationally expensive, not sufficiently accurate, or demand some a priori knowledge to meet the accuracy requirements.

This material is based on research partially sponsored by the Air Force Research Laboratory and the Defense Advanced Research Projects Agency under agreement number FA8750-16-2-0188. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Research Laboratory and the Defense Advanced Research Projects Agency or the U.S. Government.
The second author is partially supported by GNSAGA of INdAM.

In this paper we propose a novel and robust solution for PRNU pattern registration, that exploits the imaged scene content instead of relying only on PRNU information. In the general setup, a native full resolution photo and a video frame of the same static scene are firstly acquired with the device. Then, the two images are aligned with each other based on scene content via keypoint descriptor matching [9]. A refined solution minimizing small residual registration errors can then be obtained by maximizing the PRNU correlation over a limited parameter search space. Experimental evidence shows that the proposed solution is more reliable, more accurate and faster than the state-of-the-art approaches. Moreover, according to our results, PRNU pattern registration depends only on the device model for both non-stabilized videos and stabilized videos acquired with a sufficiently smooth motion. This implies that our approach can effectively be employed to build a database collecting PRNU pattern transformations for different device models, and used to develop practical applications involving a PRNU-based analysis of videos.

The rest of the paper is organized as follows. Related work is presented in Sec. 2, the proposed approach is described in Sec. 3 and experimental results are shown in Sec. 4. Conclusions and future work are discussed in Sec. 5.

2. RELATED WORK

PRNU has proved to be an effective cue for the source attribution problem in the case of digital photos [2]. Extending PRNU-based methods from photos to videos is not straightforward, and raises several issues related to the lesser reliability of videos, characterized by lower resolutions and stronger compression ratios [8]. In order to overcome these problems, PRNU-based verification usually employs only I-frames [6–8], but can also consider photo images obtained at native sensor resolution, which can be used for extracting a reference fingerprint [7]. In the latter case, an accurate alignment between the full resolution fingerprint and the video frames is required so as to compensate for the scaling and cropping of the video frame with respect to the native image. Further difficulties arise in the case of stabilized videos, since each frame is subjected to a different affine warping, aimed at alleviating the effects of camera handshaking and rolling shutter [4]. However, especially in the case of a smooth camera path, it is quite reasonable to assume that, for most part of the sequence,

stabilization is not actually enabled, thus allowing one to treat a stabilized video as a non-stabilized one [8].

Current solutions to the video PRNU pattern alignment problem search for the best PRNU pattern transformation maximizing the PRNU correlation between the fingerprint and the video frame under test, mainly focusing on translation, scale and rotation transformations. While the best translation (and then cropping) parameters can efficiently be found in the frequency domain when no other transformations are present, adding scale and rotation significantly increases the complexity of the search problem. Except for [10], the first work dealing with PRNU on stabilized videos, that anyway compensates for translation only, searching in the parameter space was usually carried out by brute-force [6, 7], and only recently replaced by particle swarm optimization as a faster and smarter search approach [8]. Nevertheless, particle swarm optimization can be still slow for some applications, and strongly depend on a sensible user-assigned parameter setup in order to work properly (for an experimental assessment of this phenomenon, see Sec. 4).

3. PROPOSED APPROACH

Our approach works with a native full resolution photo of a static scene, used as reference. A video frame captured with the same device is then acquired and aligned with the reference. When the video frame comes from a stabilized video, the reference image can be replaced by a non-stabilized video frame. As verified later in the experimental section, the transformation governing the PRNU pattern alignment depends only on the device model and not on the device exemplar at hand. This means that, once estimated under the conditions above on a given device, the same PRNU pattern transformation can be applied to all the devices sharing the same model. Figure 1 shows an example of static scene image registration. In order to improve the registration accuracy, the scene must be on focus and include discriminative patterns covering the whole image area. Only I-frames are considered and, when available, the photo and video are taken using remote or vocal controls in order to avoid any misalignments due to device motion.

For the registration, corner-like keypoints extracted with the HarrisZ detector [11] are matched with the recent SIFT-like sGLOH2 local image descriptor [12], and the initial geometric transformation parameters are estimated using RANSAC [9]. As shown in the bottom row of Fig. 1, stabilized video frames cover a smaller portion of the non-stabilized video frame area in order to compensate for affine frame deformations and avoid missing image spots from areas not covered by the camera sensor. In order to estimate the required geometric transformation, the user sets the value of the reprojection error ϵ employed to define RANSAC inliers, thus indirectly bounding the scale, rotation and translation range values. According to this observation, the transformation estimated so far can be refined through an exhaustive

search over a small set of allowable scales and rotations, operating analogously to other video PRNU pattern alignment approaches. Specifically, the PRNU correlation in terms of Peak-to-Correlation-Energy (PCE) [5] is evaluated over a small set of scales and rotations, using the PRNU fingerprint extracted from a sufficient number of images containing flat (i.e., with uniform color content) scenes of the same format as the reference image employed in the initial registration, and the PRNU extracted from a single flat video frame.

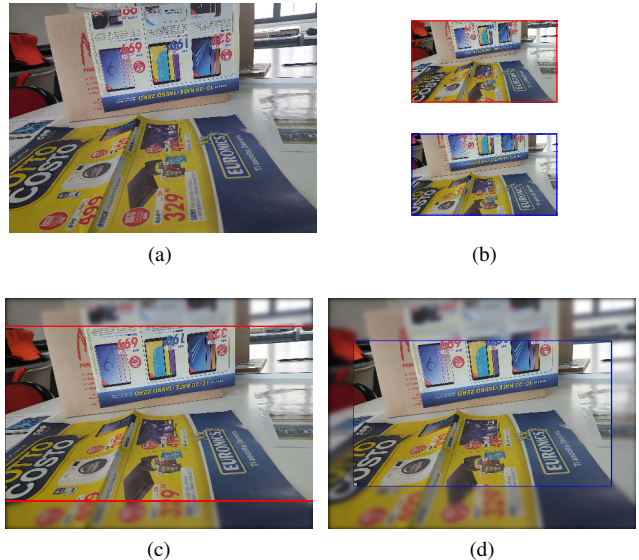


Fig. 1: Static scene image registration on a Samsung Galaxy S7 smartphone. The native full resolution photo (a) is used as reference to register the corresponding non-stabilized (b,top) and stabilized (b,bottom) video frames using local image descriptor matching. The final aligned video frames superimposed on the reference image are shown in (c) and (d), respectively. All images are scaled according to their resolution. The reference image on (c) and (d) is blurred for a better visual comparison.

In the following, details about PRNU pattern registration refinement are provided, under the assumption that rotations are negligible. This is often the case, since every good stabilization algorithm introduces only a minimal distortion, i.e., it deviates only slightly from the condition where stabilization is not actually enabled. However, the method can naturally be extended so as to take also rotations into account. A given length l on the reference image scales to $l' = l \times s$ on the video frame according to the initial scale factor s . Since the reprojection error threshold is experimentally set to $\epsilon = 4$ (i.e., the estimated average keypoint localization error), the effective scaled length l'_v ranges in the values

$$l'_v = l' + v, \quad v \in [-\epsilon, +\epsilon] \quad (1)$$

where v is quantized by a step of $q = 0.5$ pixels for computational efficiency. This leads to a set of $2\epsilon/q + 1 = 17$

allowable scale values $s_v = l'_v/l$ according to v . Considering as values for l the width and height of the reference image, and repeating the process analogously on the video frames to be evaluated, a maximum of $17 \times 4 = 68$ allowable scales s_v are required to be checked, corresponding to all the 17 possible values of v and the 4 values of l . The corresponding translation \mathbf{t}_v is obtained from s_v as

$$\mathbf{t}_v = \sum_{i=1}^n \frac{\mathbf{p}'_i - s_v \mathbf{p}_i}{n} \quad (2)$$

where $(\mathbf{p}_i, \mathbf{p}'_i)$ are the n RANSAC inlier keypoint pairs, being \mathbf{p}_i and \mathbf{p}'_i points in the reference image and in the video frame, respectively. During the scale refinement, translation values \mathbf{t}_v can be used to check the PCE peak location consistency, so as to discard the corresponding solution for relevant deviations. Note that this false alarm reduction is not possible with other approaches based only on the maximization of the PRNU correlation.

4. EXPERIMENTAL RESULTS

We compared the proposed PRNU pattern registration approach on seven different devices against particle swarm optimization, which provides better accuracy and computational efficiency than brute-force approaches. In particular, for each device, we registered in terms of scale and translation video I-frames with a flat homogeneous scene content over PRNU fingerprints extracted from native resolution photo images or I-frames of non-stabilized videos. Following the assumption of the previous section, no rotations were taken into account. Table 1 reports the results in terms of PCE and the running time, together with the estimated scale. Due to lack of space, only the mean μ , standard deviation σ , and minimum and maximum values are reported here (detailed results, dataset and code are provided as additional material for further analysis and reproducibility¹). The scene-based registration and its refinement are indicated as G and G_r , respectively. Additionally, G_m represents the results obtained by averaging G_r scales while discarding video I-frames with low PCE values (i.e., less than 50) on G , as a fast way to skip unusable frames. For particle swarm, implemented using the Matlab particle swarm function, two different versions are provided according to two different initial parameter setups. In detail, setup P uses 35 particles and a scale search range in $[0.5, 3]$, while setup P_r uses 30 particles and a scale search range in $[1, 3]$ and $[0.5, 1]$, respectively when the reference PRNU is extracted from native full resolution photos and non-stabilized videos. Notice that the total running time for G_r is obtained by adding the corresponding columns G and G_r in the table.

The mean PCE value obtained with the scene content registration method G only is in most cases quite accurate, even without scale refinement (method G_r). The average registration G_m gives values very close to those given by G_r . The

almost identical scale values obtained with the two different Samsung S7 devices witness the fact that PRNU pattern registration is not device dependent and, hence, G_m registration can be used with other devices of the same model, thus avoiding to acquire each time ad hoc static scene images or videos. Moreover, as reported in the additional material, registration of native full resolution images onto non-stabilized and stabilized video frames can be concatenated to get a registration from non-stabilized to stabilized videos without any accuracy degradation. Concerning particle swarm optimization, the solution obtained with P_r is usually more accurate and reliable than with P , confirming the fact anticipated above that, without any a priori clues, a sensible scale range must be provided by the user in order to avoid unstable or even wrong solutions. Scene-based PRNU pattern registration is in general more accurate and reliable than that obtained by particle swarm optimization. Only for the Huawei P9 Lite and, to a lesser extent, for the 1st Samsung Galaxy S7 device when registering non-stabilized videos over native full resolution photos, our approach tends to be less accurate, probably due to a noisy acquisition of the static scene images and videos when preparing the data.

Concerning running times, scene-based registration G is very fast and even by summing up the further refinement step G_r , the approach is faster than particle swarm optimization. In particular, our full approach G_r is about four times faster than particle swarm optimization, except for non-stabilized to stabilized video PRNU registration with setup P_r , for which our approach is only twice faster. Note that running times depend on image resolution and hence on the scale search range. Clearly, particle swarm accuracy can be improved by using more particles in the setup, yet computation time would increase accordingly.

5. CONCLUSIONS AND FUTURE WORK



























This paper proposed a novel and robust solution for PRNU pattern alignment, that uses the image scene content to obtain an initial registration using local image descriptors and can be further refined by maximizing the PRNU correlation. The proposed approach has shown to be more reliable, more accurate and faster than existing approaches based on brute-force and particle swarm optimization. Future work will include the integration of the proposed scene content registration into the particle swarm framework, as a tool to drive the search in the parameter space. Additionally, we plan to extend the method to include rotations that are reasonably expected on long-time non-flat stabilized videos, currently excluded in our analysis. Finally, a more accurate acquisition protocol for the static scene will be defined, concurrently with the generation of a shared database collecting the estimated PRNU transformation for several device models.

6. ACKNOWLEDGMENTS

We heartily thank the authors of [8] for sharing their code.

¹ <https://drive.google.com/open?id=1hfqgWDBZRxTErDNAjQg-GtTYUj267gd1>

Table 1: PRNU registration evaluation results (see text for details).

Device model	Registration mode	Scale					PCE					Time (sec)				
		G	G_r	G_m	P	P_r	G	G_r	G_m	P	P_r	G	G_r	P	P_r	
Samsung Galaxy A3	 → 	μ	1.6993	1.7001	1.7001	2.3125	2.2454	5826	7746	7746	1691	1358	18	44	524	439
		σ	-	0.0000	-	0.6059	0.5941	1426	1922	1922	2289	2191	-	0	100	82
		min	-	1.7001	-	1.6976	1.6967	656	820	820	42	42	-	43	387	341
		max	-	1.7001	-	2.9920	3.0000	6905	9176	9176	6433	8627	-	45	658	606
Huawei P9 Lite	 → 	μ	0.7944	0.7951	0.7981	1.5721	0.7726	107	220	183	612	1019	8	14	343	45
		σ	-	0.0022	-	0.9340	0.1370	83	314	309	1308	2122	-	0	165	1
		min	-	0.7912	-	0.5000	0.5000	37	50	31	67	71	-	13	81	43
		max	-	0.7981	-	3.0000	1.0000	523	1535	1535	6656	8728	-	14	686	46
Samsung Galaxy S7 (1 st device)	 → 	μ	2.0924	2.0966	2.0982	1.2509	1.8340	31	488	486	381	603	39	83	359	347
		σ	-	0.0029	-	0.7366	0.5014	5	360	362	522	527	-	0	28	32
		min	-	2.0895	-	0.5000	1.0000	26	32	26	39	38	-	82	326	285
		max	-	2.0982	-	2.1046	2.5687	49	925	925	1674	1559	-	84	399	393
	 → 	μ	1.7512	1.7515	1.7499	1.1065	1.6360	104	125	105	79	93	38	81	364	341
		σ	-	0.0026	-	0.5700	0.5861	93	98	92	50	78	-	0	17	53
		min	-	1.7471	-	0.5086	1.0000	28	32	27	45	35	-	81	338	281
		max	-	1.7552	-	1.7571	3.0000	409	431	361	226	355	-	82	399	522
 → 	μ	0.8372	0.8356	0.8344	2.2775	0.7446	60	152	130	65	123	9	14	489	44	
	σ	-	0.0018	-	0.8528	0.1331	34	130	129	53	96	-	0	132	1	
	min	-	0.8335	-	0.5060	0.5000	26	32	23	38	38	-	14	151	44	
	max	-	0.8405	-	3.0000	0.8374	129	534	534	321	414	-	15	705	45	
Samsung Galaxy S7 (2 nd device)	 → 	μ	2.1000	2.0997	2.0997	1.7740	2.1976	1168	1233	1233	456	521	34	73	411	404
		σ	-	0.0000	-	0.7310	0.4968	298	313	313	473	527	-	1	63	72
		min	-	2.0997	-	0.5000	1.0000	434	478	478	37	37	-	73	329	291
		max	-	2.0997	-	3.0000	3.0000	1820	1920	1920	1453	1701	-	77	674	604
	 → 	μ	1.7485	1.7494	1.7499	1.3374	2.0793	212	347	336	137	117	33	81	384	399
		σ	-	0.0015	-	0.7728	0.6922	237	429	429	248	130	-	0	59	73
		min	-	1.7448	-	0.5000	1.0051	27	31	25	37	37	-	80	325	289
		max	-	1.7526	-	3.0000	3.0000	1255	2249	2249	1457	563	-	82	658	580
 → 	μ	0.8325	0.8332	0.8333	2.5241	0.7762	1461	2700	2620	319	1648	8	14	552	44	
	σ	-	0.0012	-	0.7787	0.1147	1533	2980	2945	851	1875	-	0	120	1	
	min	-	0.8293	-	0.8297	0.5000	23	29	22	47	34	-	13	261	43	
	max	-	0.8363	-	3.0000	0.9720	6749	12531	12392	4182	6004	-	14	669	48	
Sony Xperia XA1 G3112	 → 	μ	2.8777	2.8782	2.8759	1.5447	2.2467	95	134	129	94	86	72	182	834	719
		σ	-	0.0035	-	0.9944	0.6782	131	203	204	152	112	-	1	21	21
		min	-	2.8725	-	0.5000	1.0782	28	33	27	36	37	-	181	811	700
		max	-	2.8857	-	3.0000	2.9949	452	674	674	665	552	-	183	906	788
	 → 	μ	2.3013	2.3005	2.3003	1.2127	1.6330	38	43	38	49	48	70	200	824	711
		σ	-	0.0019	-	0.6753	0.7032	14	15	16	19	17	-	1	8	15
		min	-	2.2962	-	0.5000	1.0000	27	32	26	37	36	-	199	811	696
		max	-	2.3050	-	2.8759	3.0000	93	98	93	161	139	-	202	840	766
 → 	μ	0.7998	0.7997	0.8001	2.5396	0.8114	784	1119	860	443	880	9	14	540	45	
	σ	-	0.0017	-	0.6737	0.1185	839	1205	951	453	919	-	0	108	0	
	min	-	0.7961	-	0.5104	0.5085	58	84	58	185	102	-	13	265	44	
	max	-	0.8035	-	3.0000	0.9984	2534	3475	2632	3143	3161	-	15	712	45	
iPhone 4S	 → 	μ	1.3343	1.3335	1.3334	1.4217	1.7556	2974	4383	4081	1441	1852	25	57	359	362
		σ	-	0.0008	-	0.6236	0.6985	1614	2485	2298	1982	2425	-	0	96	90
		min	-	1.3327	-	0.5003	1.1685	253	453	174	47	41	-	57	259	227
		max	-	1.3365	-	2.9918	3.0000	5928	8361	8212	6910	7341	-	58	594	522
iPhone 6S	 → 	μ	1.7754	1.7772	1.7778	1.2848	1.6941	1127	1800	1767	927	1314	36	81	357	315
		σ	-	0.0015	-	0.5712	0.3217	520	918	921	957	925	-	0	17	23
		min	-	1.7723	-	0.5000	1.0000	29	33	28	44	40	-	80	331	284
		max	-	1.7782	-	1.7812	2.2073	1844	3133	3105	2860	2844	-	81	378	365

The "registration mode" column indicates which image formats are employed for the registration, the reference format being on left.

 photo  unknown mode video  non-stabilized video  stabilized video

7. REFERENCES

- [1] J. Lukas, J. Fridrich, and M. Goljan, "Digital camera identification from sensor pattern noise," *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 2, pp. 205–214, 2006.
- [2] M. Chen, J. Fridrich, M. Goljan, and J. Lukas, "Determining image origin and integrity using sensor noise," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 1, pp. 74–90, 2008.
- [3] D. Shullani, M. Fontani, M. Iuliani, O. Al Shaya, and A. Piva, "VISION: a video and image dataset for source identification," *EURASIP Journal on Information Security*, vol. 2017, no. 1, pp. 15, 2017.
- [4] M. Grundmann, V. Kwatra, and I. Essa, "Auto-directed video stabilization with robust 11 optimal camera paths," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 225–232.
- [5] M. Goljan and J. J. Fridrich, "Camera identification from cropped and scaled images," in *Proceeding of Security, Forensics, Steganography, and Watermarking of Multimedia Contents X*, 2008.
- [6] S. Taspinar, M. Mohanty, and N. Memon, "Source camera attribution using stabilized video," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2016, pp. 1–6.
- [7] M. Iuliani, M. Fontani, D. Shullani, and A. Piva, "Hybrid reference-based video source identification," *Sensors*, vol. 19, no. 3, 2019.
- [8] S. Mandelli, P. Bestagini, L. Verdoliva, and S. Tubaro, "Facing device attribution problem for stabilized video sequences," *arXiv preprint arXiv:1811.01820*, 2018.
- [9] R. Szeliski, *Computer Vision: Algorithms and Applications*, Springer-Verlag, 1st edition, 2010.
- [10] T. Höglund, P. Brolund, and K. Norell, "Identifying camcorders using noise patterns from video clips recorded with image stabilization," in *Proceedings of the International Symposium on Image and Signal Processing and Analysis (ISPA)*, 2011, pp. 668–671.
- [11] F. Bellavia, D. Tegolo, and C. Valenti, "Improving Harris corner selection strategy," *IET Computer Vision*, vol. 5, no. 2, pp. 86–96, 2011.
- [12] F. Bellavia and C. Colombo, "Rethinking the sGLOH descriptor," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 931–944, 2018.