HAL
open science

Countering Noise-based Splicing Detection Using Noise Density Transfer

Thibault Julliand, Vincent Nozick, Hugues Talbot

To cite this version:

Thibault Julliand, Vincent Nozick, Hugues Talbot. Countering Noise-based Splicing Detection Using Noise Density Transfer. Journal of Digital Forensics, Security and Law, 2016, 11 (2), pp.111-122. 10.15394/jdfsl.2016.1382 . hal-01510073

HAL Id: hal-01510073

https://hal.science/hal-01510073v1

Submitted on 19 Apr 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

COUNTERING NOISE-BASED SPLICING DETECTION USING NOISE DENSITY TRANSFER

Thibault Julliand†, Vincent Nozick* and Hugues Talbot†
Université Paris-Est, LIGM (UMR 8049), CNRS, ENPC, ESIEE Paris, UPEM,
†F-93162, Noisy-le-Grand, France *F-77454, Marne-la-Vallée, France
{thibault.julliand, hugues.talbot}@esiee.fr, vincent.nozick@u-pem.fr

ABSTRACT

Image splicing is a common and widespread type of manipulation, which is defined as pasting a portion of an image onto a second image. Several forensic methods have been developed to detect splicing, using various image properties. Some of these methods exploit the noise statistics of the image to try and find discrepancies. In this paper, we propose a new counter-forensic approach to eliminate the noise differences that can appear in a spliced image. This approach can also be used when creating computer graphics images, in order to endow them with a realistic noise. This is performed by changing the noise statistics of the spliced elements so that they are closer to those of the original image. The proposed method makes use of a novel way to transfer density functions. We apply this to image noise in order to impose identical noise density functions from a source to a destination image. This method can be used with arbitrary noise distributions. The method is tested against several noise-based splicing detection methods, in order to prove its efficacy.

Keywords: counter-forensics, splicing, image forgery, distribution transfer

1. INTRODUCTION

Over the past ten years, digital images have become an ever-growing element of our daily lives. They are used nowadays in a wide range of situations: medias, social networks, justice, surveillance, but also as personal items, for example on smartphones. Additionally, software that can be used to alter images have become both more common and easier to use, with Photoshop being among the most well-known. Consequently, the falsification of digital images has also become more frequent and easy. Falsified images can be used in almost any of the setting evoked before, for example to obfuscate elements of a photograph or in slander cases.

One of the most widespread falsification is the exogenous insertion, colloquially known as splicing. Splicing consists of adding elements from one or multiple images in an original image, giving a final result that is the composite of different images. However, several methods have emerged to detect alterations in digital images, including splicing, referred to as the field of forensic imagery. As those methods are based on studying properties of altered images, a new, concurrent field has appeared, dedicated to countering the forensics methods: anti-forensics. The main idea behind anti-forensics is to find ways to hide the detectable signs of an alteration, without degrading its quality.

Another face of digital imaging are artificial images generated using computer graphics, referred to hereafter as CG images. The quality of CG images has drastically increased over the past few years, to the point that some of them can fool a human observer. Although they are commonly used in video games or movies, CG images can have more ill-intentioned uses. They are, for example, used in cases concerning

identity theft, usually combined with a voice-alteration software. Although high-quality CG images can deceive the human eye, there are some physical or statistical properties that can be used by forensic methods to differentiate them from natural images. Consequently, as with natural images, some counter-forensics methods have emerged in order to hide those properties.

In this paper, we present a novel way to transfer the noise from one image to another, and two different uses for it: first, conceal a spliced element in a natural image. Second, make a CG image appear more natural both from the human perspective and against forensics software.

2. NOISE DENSITY TRANSFER

A first naive approach to transfer noise from one image to another would be to denoise both images, obtain the noise image (difference between the original and denoised image) of the first one, and add it onto the second one. This, of course, causes several problems: the size of the two images has to be the same and the contours of the first image will be superimposed on the second one, due to the fact that all denoising methods leave traces correlated to the contours of the denoised image.

Another method would consist of estimating the noise variance in the first image and applying an artificial Gaussian noise with the same variance on the second image after denoising. Although this would produce a reasonably convincing result, at least for human observers, a perfect additive Gaussian white noise is never found in natural images. Indeed, Julliand et al. (Julliand, Nozick, & Talbot, 2015) show that though a standard digital image noise can be approximated by a Gaussian distribution, it is never perfect. Additionally, the denoising step will clearly leave some traces, which could be detected with adequate methods, such as the blur detection method presented in (Zhou, Wang, Guo, & Zhang, 2007).

2.1 State of the art - Density Transfer

The probability density function associated to a random variable, such as noise in an image, describes the odds for this random variable to have a given value. The method we propose is based on transforming the noise density function of a first image so that it is as close as possible to that of another image. The problem of transferring and changing a density function has already been studied extensively in several optimization problems. The concept was first introduced by Monge in 1791(Monge, 1781), and more recently studied statisticians in the 1970s (Vallender, 1974) under the moniker of Wasserstein distances. It was applied to images by Peleg et al. in (Peleg, Werman, & Rom, 1989), where it is used to propose a method to change the resolution of an image while avoiding aliasing. Several optimized versions of the generic transfer function have been developed by Xu et al. (Xu et al., 2015) and Huang et al. (Huang, Zhang, Buyya, & Chen, 2015). However, the purpose of our approach is not only to match one probability distribution to another, but also to get a credible image noise as a result. Additionally, both of the optimized methods are used to process very large amounts of data, while in our case the quantity of changes to be done in a distribution are limited by the size of the image. As a consequence, we do not need to focus on the optimal number of operations to transform the distributions. The method may be implemented in a faster and more efficient way, but this would not affect the final result.

2.2 Noise density table

Noise in digital images may have various sources. In a raw image, noise follows a Poisson-Gauss probability distribution, with the standard deviation varying with the intensity of each pixel. The exact function defining the Poisson-Gauss probability density table is shown in Eq.1, where σ is the standard deviation of the Gaussian portion of the function and α a scaling parameter

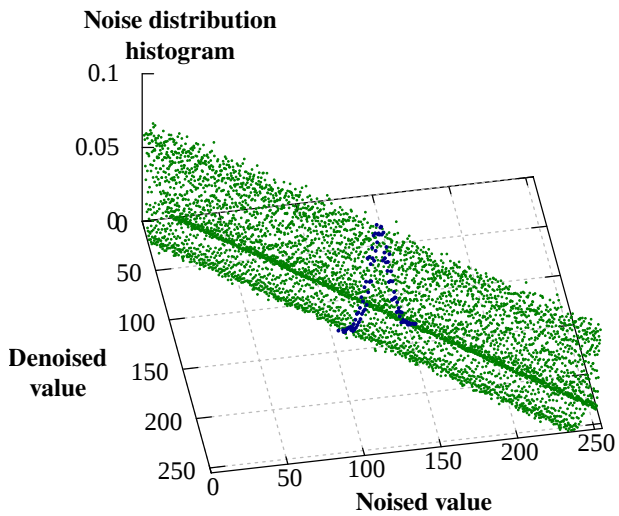


Figure 1: Gaussian density table. The blue line is a cross-section along a single denoised value.

applied to the Poisson portion:

$$f(v_d, v_n) = \frac{\alpha}{\sigma\sqrt{2\pi}} \sum_{x=0}^{\infty} \frac{(\alpha v_d)^{\alpha x} e^{-\alpha v_d}}{(\alpha x)!} \exp\left(\frac{-(v_n - x)^2}{2\sigma^2}\right) \quad (1)$$

In JPEG images, noise can be accurately modeled as an additive white Gaussian noise. Eq. 2 shows the probability density function for a pure Gaussian noise.

$$f(v_d, v_n) = \frac{1}{\sigma\sqrt{2\pi}} exp\left(\frac{-(v_n - v_d)^2}{2\sigma^2}\right) \quad (2)$$

A noise density table is the representation of this probability distribution. For each pixel, we consider its denoised value v_d and its noisy value v_n. To each pixel of the image corresponds a value pair (v_d, v_n), which are accumulated in the table. This way, the table can be seen as a 2D histogram, as depicted in Fig 1.

In practice, the value in the table at any point (i, j) is the amount of value pairs where $(v_d, v_n) = (i, j)$. For numerical purposes, the table is normalized on each row (denoised values) to offset potential intensity imbalances in the image. Indeed, the table of an image with a large proportion of high (or low) intensity pixels would have very high values in the corresponding areas. This would reduce the usability of the table. The normalization solves this problem.

2.3 Noise transfer

The proposed approach is based on transferring the noise density table of an image onto another, in contrast to just generating a Gaussian noise with the correct standard deviation on the second image. The distinction between these two techniques is important: indeed, the proposed method can copy any kind of noise density table, and as such is not only limited to images with Gaussian noise.

We note I_s the image source of the transfer and I_d the image destination of the transfer. The first step is to denoise these two images, giving us the denoised images $\widehat{I_s}$ and $\widehat{I_d}$. Note that these denoised images $\widehat{I_s}$ and $\widehat{I_d}$ are used only for the density tables computation, but are not used in the noise transfer itself. Finally, we compute the corresponding density tables T_s and T_d. See Fig. 2.

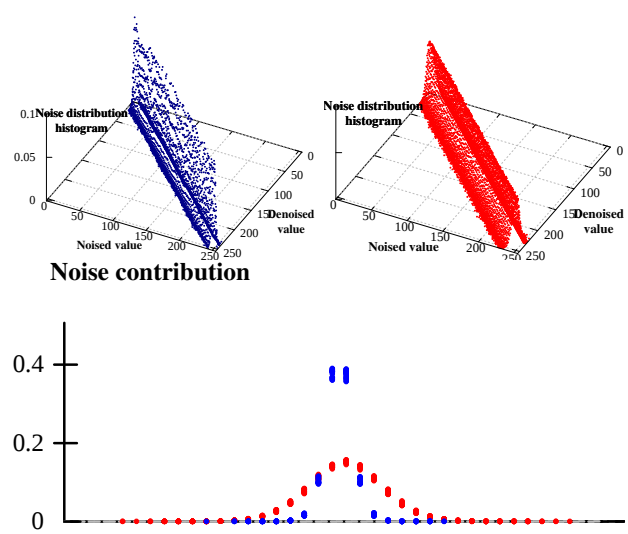


Figure 2: 2.3 and 2.3 are the density tables of the source and destination images respectively. 2.3 shows a projection of those two map along the identity axis for easier comparison.

For the transfer, we consider that a density table is bounded as follow: each row represents

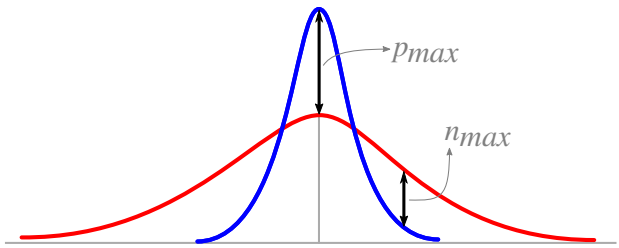


Figure 3: A representation of the maximal differences and their location. The curves represent a cut of the density function along a denoised value.

the possible values of the denoised image, and each column represents the possible values of the noised image. Both rows and columns can take a value ranging in $1...N$. For example, $N = 255$ for images with 8-bits per channel.

On a given row r, we look for the two maximal differences column indexes c_p and c_n. Theses indexes are formally defined as:

$$c_p | \Big(T_s(r, c_p) - T_d(r, c_p)\Big) = \max_{1 \leq i \leq N} \Big(T_s(r, i) - T_d(r, i)\Big)$$

$$c_n | \Big(T_d(r, c_n) - T_s(r, c_n)\Big) = \max_{1 \leq j \leq N} \Big(T_d(r, j) - T_s(r, j)\Big)$$

We then define the corresponding maximal differences (see Fig.3) as:

$$p_{max} = \Big(T_s(r, c_p) - T_d(r, c_p)\Big)$$

$$n_{max} = \Big(T_d(r, c_n) - T_s(r, c_n)\Big)$$

Once the maximal differences and their positions are found, we change the value of a random pixel on the destination image with a noised value of c_n and a denoised value of r to a noised value of c_p. We then update the destination density table to reflect the change. This will have the effect of altering the shape of the density table T_d and bring it closer to T_s, by reducing both p_{max} and n_{max}. This process is iterated on the column until convergence. We consider convergence is reached when the destination density table takes the same configuration two times in row. Once convergence is reached for a row, we process the next one until all rows are transferred, as shown in Fig. 4.

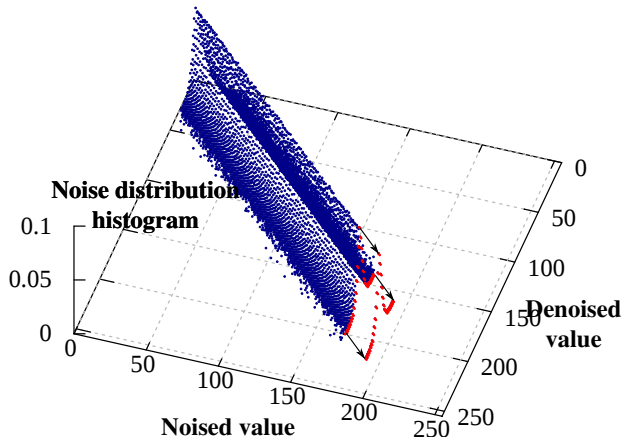


Figure 5: A representation of the gamut management. The red curve is translated and copied along the identity axis to fill the row where it is needed.

2.4 Gamut management

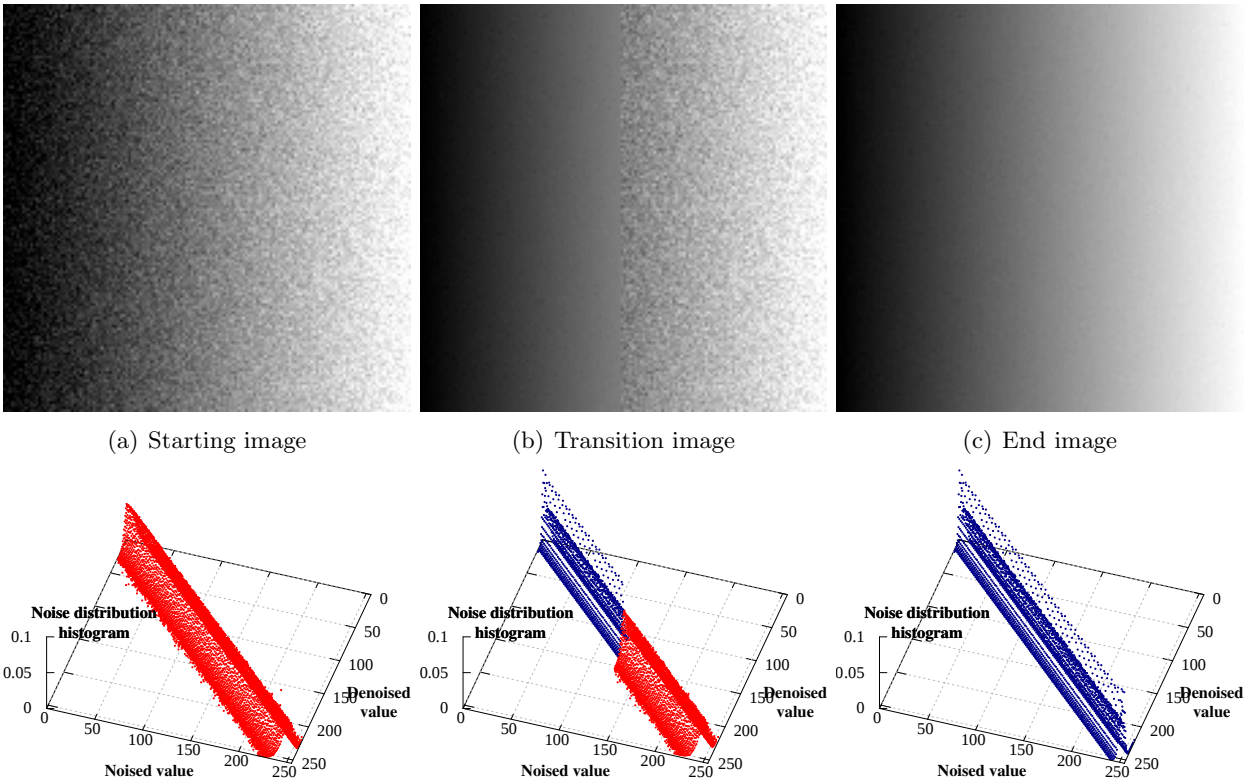
In the case where the destination table has a row which is empty on the source table, it is necessary to fill the row with coherent data for the sake of the transfer. In the case of a Gaussian distribution, we simply translate the closest non-empty row along the identity axis, as shown in Fig. 5. In the case of a Poisson-Gauss distribution, this may not work if the closest non-empty row is too far away and as such has a clearly different standard deviation. In that case, if we assume that the noise is independent in each channel, we can look in each of the other channels for a suitable replacement.

3. APPLICATION: SPLICING CAMOUFLAGE

Splicing is an extremely common form of image alteration. A splicing consists in inserting part of an image in a different image.

3.1 State of the art

Several ways to detect splicing already exist, using various properties of an image: (Popescu & Farid, 2005) exploits the perturbations of the Color Filter Array, (Farid, 2009) uses JPEG ghosting, (Lin, He, Tang, & Tang, 2009) is



(a) Starting image (b) Transition image (c) End image

(d) Starting density table (e) Transition density table (f) End density table

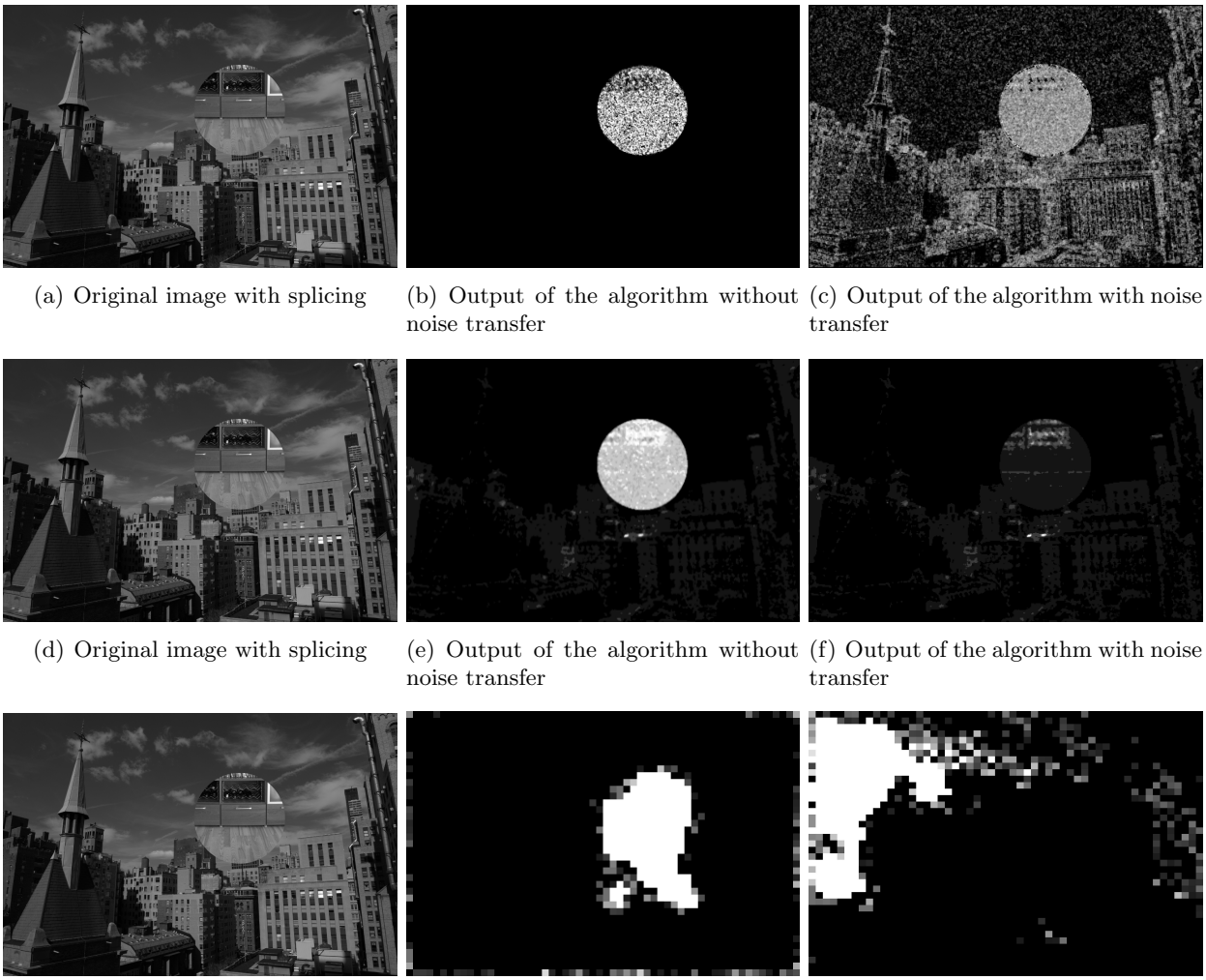
Figure 4: Images in the top row show the impact of the noise transfer on an image. The noise has been voluntarily amplified so that the results are more easily visibles. The bottom row show the evolution of the density table through the transfer.

based on analysing the quantization in JPEG images, and (Wang, Dong, & Tan, 2009) uses the image color gamut associated with machine learning. It is important to note that most of those methods are strictly restricted to JPEG images.

A constant element in all natural digital images is the noise. This noise is created by multiple sources, and may be reduced by various factors, such as the camera post-processing pipeline or a strong compression, as shown in (Julliand et al., 2015). However, the parameters of the remaining noise will be consistent throughout the image. Several splicing detection methods are based on detecting local inconsistencies in those parameters, which can be a strong indicator of splicing in an image.

Mahdian and Saic (Mahdian & Saic, 2009) offer a block-based approach relying on analysing the standard deviation of the noise throughout the image. The noise is approximated from a first order wavelet decomposition. This method relies on the assumption that the noise is Gaussian, and as such can not be used on several image formats. Pan et al. (Pan, Zhang, & Lyu, 2012) also propose a block-based approach and consider that kurtosis values of natural images in band-pass filtered domains tend to concentrate around a constant value. This kurtosis estimation is used to compute a local noise variance estimation, and extract inconsistencies. Finally, Julliand et al. (Julliand, Nozick, & Talbot, 2016) propose a method using local noise density function discrepancies to expose splicing.

Several methods have emerged in recent years to counter the development of digital forensics. When it comes to conceal splicings, most of



(a) Original image with splicing (b) Output of the algorithm without noise transfer (c) Output of the algorithm with noise transfer

(d) Original image with splicing (e) Output of the algorithm without noise transfer (f) Output of the algorithm with noise transfer

(g) Original image with splicing (h) Output of the algorithm without noise transfer (i) Output of the algorithm with noise transfer

Figure 6: The effect of our method on three detection approaches. 6(a), 6(b) and 6(c) show the effect on the approach designed by Pan et al. ; 6(d), 6(e) and 6(f) use the approach of Mahdian et al., and 6(g), 6(h) and 6(i) the approach of Julliand et al. In all of the cases, the spliced element cannot be distinguished from the rest of the image after the noise transfer.

the anti-forensics approaches are focused on the JPEG format. In (Kirchner & Bohme, 2007) and (Kirchner & Bohme, 2009), the authors present counters to hide several alterations, like CFA disturbance or image resampling. Likewise, (Stamm & Liu, 2011) propose to alter the transform coefficients of an image during compression, to hide the impact of several falsifications like copy-move, splicing, or double-JPEG on the compression history of an image. However, to our knowledge, there are no existing methods to dissimulate the impact of splicing on the noise in an image.

3.2 Test protocol

The first step of our experiments were to get an estimation of the standard deviation of the noise on the images, then use it to denoise them. The noise estimation was made using Colom and Buades work (Colom & Buades, 2014), and denoising was performed using the Lebrun (Lebrun, 2012) implementation of BM3D.

We tested the efficiency of our approach against three different splicing detection algorithm based on detecting noise inconsistencies: (icip16), (Pan et al., 2012), and (Mahdian & Saic, 2009). Our protocol was the same in the three cases: we splice two images without altering the noise on any of them. Then, we analyse the spliced image with the detection algorithm. Finally, we repeat this procedure, but we use our noise transfer approach before the splicing. This allows us to highlight the impact of our method.

3.3 Results

On both Fig. 6 and Fig. 7, we can see that after being subjected to our method, the spliced elements can not be found in the images, and this against the three detection approach we used. Indeed, in the case of Pan et al. and Mahdian et al., the spliced element noise is level with the rest of the image, and thus indistinguishable. In the case of Julliand et al., the algorithm does seem to find that part of the image have been altered, but not in the correct position at all.

3.4 Consequences and discussion

As discussed in section 2.1, our method presents several advantages over other more naive methods. The most common naive approach is to denoise the image whose noise we want to change, and renoise it with an artificial Gaussian white noise. However, this presents several issues: first, not all images have a noise distribution that follows a Gaussian distribution. Some noises follow a Poisson distribution, or a Poisson-Gauss one. In this last case in particular, getting the parameters of the noise to reproduce it is considerably harder than with a simple Gaussian noise. Second, this method needs a denoising step, and add the noise on the denoised image. As there is no perfect denoising algorithm, the denoised image will necessarily have denoising artefacts which can take different forms, the most common being a slight blur or areas around edges that are not perfectly denoised. To our knowledge, no method exists to detect this second case, but some algorithms can already detect blurring in an image, even if it is renoised, as the one presented in (Zhou et al., 2007) and (Cao, Zhao, & Ni, 2010).

In comparison, our method only uses denoising in order to compute the density tables. All of the image modifications are made directly on the noised image. Furthermore, the fact that we select the pixel to change randomly among those eligible guarantees that there will be no observable blur or added noise around the edges.

In addition, our approach can be combined with other anti-forensics methods to hide the possible other effects of a splicing, such as the alteration of the CFA pattern.

4. APPLICATION: CG CAMOUFLAGE

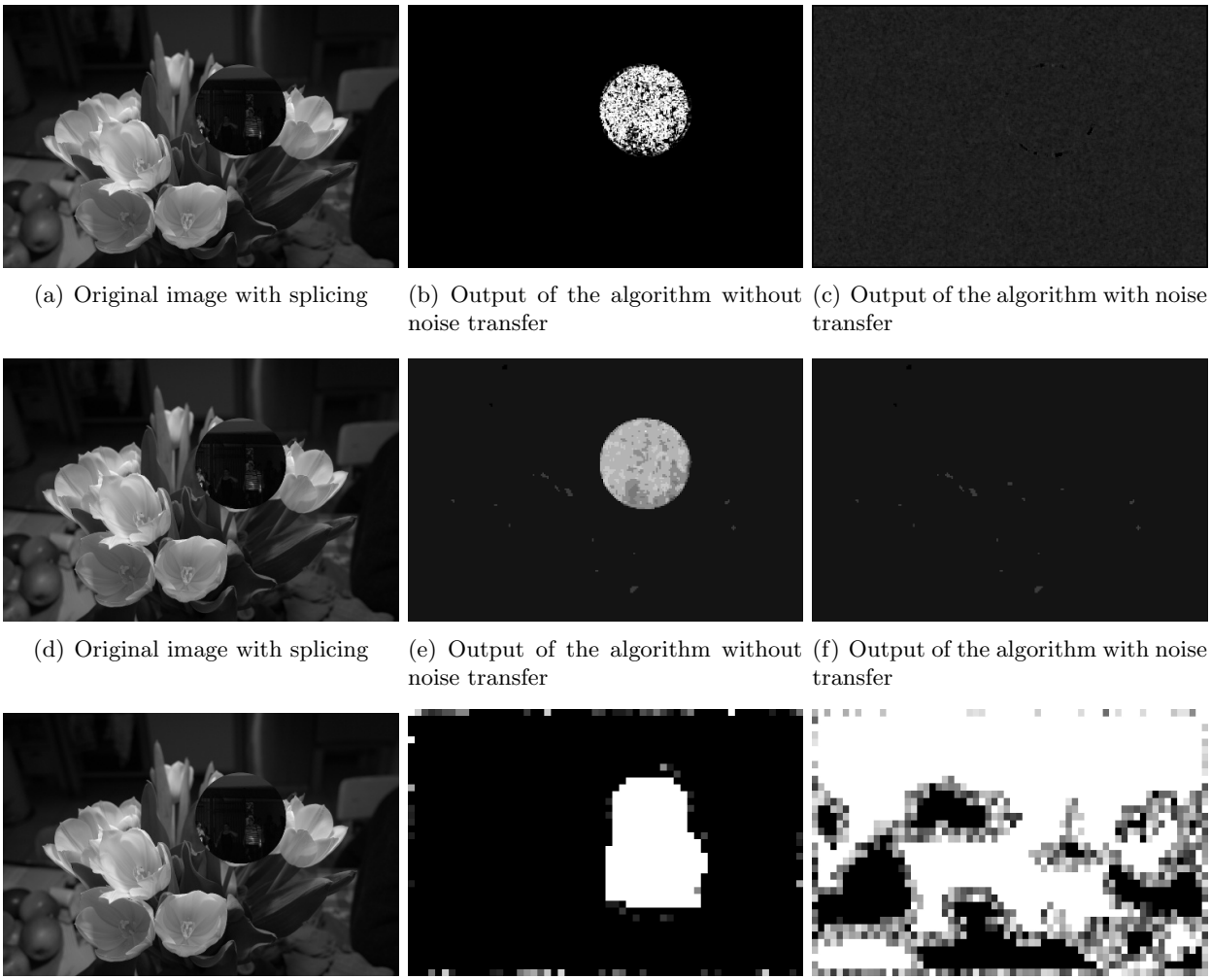
4.1 State of the art

According to a study by Fan et al. (Fan, Ng, Herberg, Koenig, & Xin, 2012), image noise is not an element that helps human to distinguish a CG from an natural image. However, several forensics methods have been developed to make this distinction.

The first relevant method to distinguish CG from natural images was introduced by Farid and Lyu (Farid & Lyu, 2003). They extract some features from a separable quadrature mirror filters applied on the image, and use a learning framework to classify CG from natural image. From then, nearly all the methods follow the same approach, proposing more relevant features vectors.

Among these contributions, Chen et al. (Chen, Shi, & Xuan, 2007) explored the feature extraction on HSV color space whereas concurrence only deals with each RGB channels independently.

Wu et al. (Wu, Li, & Yang, 2011) simplify the usual wavelet-based features by just using



(a) Original image with splicing (b) Output of the algorithm without noise transfer (c) Output of the algorithm with noise transfer

(d) Original image with splicing (e) Output of the algorithm without noise transfer (f) Output of the algorithm with noise transfer

(g) Original image with splicing (h) Output of the algorithm without noise transfer (i) Output of the algorithm with noise transfer

Figure 7: The results on this set of images are similar to the one in Fig. 6.

histograms of some measures on these wavelet decomposition. This method is one of the most accurate in the state of the art and is also the easiest to implement.

For more details, the reader can refer to Ng and Chan (Ng & Chang, 2013) as well as Tokuda et al. (Tokuda, Pedrini, & Rocha, 2013) who present some interesting performances evaluation tables comparing a large set of state of the art methods.

4.2 Consequences and discussion

In contrast to adding noise to natural images, which requires pre-emptive denoising, adding noise to CG images can be done without a denoising step, depending on the way the image was generated (some ray-tracer methods, like bidirectional path tracing, induce noise in the final image). As a consequence, there is no risk of denoising artefacts. However, this also guarantees that the resulting image will have a perfect additive white Gaussian noise, which is not found in natural images, except when very strong compression is involved. As seen in section 4.1, forensics methods able to differentiate

CG and natural images are based on first order wavelet analysis (or similar alternatives) that is highly related to image noise.

These detection strategies exploit the unnaturalness of this perfect noise. Since our method can transfer the noise from a natural image instead of adding a generated noise, there is, in theory, a reduced chance to be detected by such algorithms. However, it is important to note that those approaches use other features beyond the pure noise statistics, and as such the impact of a more natural noise requires more research.

5. CONCLUSION

In this article, we have proposed a novel way to alter the noise of an image by transferring the noise density from another image. We have also presented two possible applications for this method. The first application is to dissimulate image splicing. The method is adaptable to various type of noises, and is able to fool several state-of-the-art noise-based splicing detection algorithms. The second possible application is to make computer-generated images appear less artificial, by giving them the noise of a natural image.

REFERENCES

Cao, G., Zhao, Y., & Ni, R. (2010). Edge-based blur metric for tamper detection. *Journal of Information Hiding and Multimedia Signal Processing*.

Chen, W., Shi, Y. Q., & Xuan, G. (2007). Identifying computer graphics using hsv color model and statistical moments of characteristic functions. In *2007 ieee international conference on multimedia and expo* (pp. 1123–1126).

Colom, M., & Buades, A. (2014). Analysis and extension of the pca method, estimating a noise curve from a single image. *Image Processing On Line*.

Fan, S., Ng, T.-T., Herberg, J. S., Koenig, B. L., & Xin, S. (2012). Real or fake?: human judgments about photographs and computer-generated images of faces. In *Siggraph asia 2012 technical briefs* (p. 17).

Farid, H. (2009). Exposing digital forgeries from jpeg ghosts. *IEEE Transactions on Information Forensics and Security*.

Farid, H., & Lyu, S. (2003). Higher-order wavelet statistics and their application to digital forensics. In *Ieee workshop on statisticalanalysis in computer vision* (p. 94). Madison, Wisconsi.

Huang, J., Zhang, R., Buyya, R., & Chen, J. (2015). Heads-join: Efficient earth movers distance join on hadoop. *IEEE Transactions on Parallel and Distributed Systems*.

Julliand, T., Nozick, V., & Talbot, H. (2015). Image noise and digital image forensics. *Digital Forensics and Watermarking*.

Julliand, T., Nozick, V., & Talbot, H. (2016, February). *Automatic image splicing detection based on noise density analysis in raw images.* (working paper or preprint)

Kirchner, M., & Bohme, R. (2007). Tamper hiding: Defeating image forensics. *Information Hiding*.

Kirchner, M., & Bohme, R. (2009). Synthesis of color filter array pattern in digital images. *Media Forensics and Security*.

Lebrun, M. (2012). An analysis and implementation of the bm3d image denoising method. *Image Processing On Line*, *2*, 175–213.

Lin, Z., He, J., Tang, X., & Tang, C.-K. (2009). Fast, automatic and fine-grained tampered jpeg images detection via dct coefficient analysis. *Pattern Recognition*.

Mahdian, B., & Saic, S. (2009). Using noise inconsistencies for blind image forensics. *Image and Vision Computing*.

Monge, G. (1781). Mémoire sur la théorie des déblais et des remblais. *Histoire de l'Académie Royale des Sciences de Paris, avec les Mémoires de Mathématique et de Physique pour la même année*, 666–704.

Ng, T.-T., & Chang, S.-F. (2013). Discrimination of computer synthesized or recaptured images from real images. In

Digital image forensics (pp. 275–309). Springer.

Pan, X., Zhang, X., & Lyu, S. (2012). Exposing image splicing with inconsistent local noise variances. *International Conference on Computation Photography (ICCP)*.

Peleg, S., Werman, M., & Rom, H. (1989). A unified approach to the change of resolution: Space and gray-level. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Popescu, A. C., & Farid, H. (2005). Exposing digital forgeries in color filter array interpolated images. *IEEE Transactions on Signal Processing*.

Stamm, M. C., & Liu, K. J. R. (2011). Anti-forensics of digital image compression. *IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY*.

Tokuda, E., Pedrini, H., & Rocha, A. (2013). Computer generated images vs. digital photographs: A synergetic feature and classifier combination approach. *Journal of Visual Communication and Image Representation*, *24*(8), 1276–1292.

Vallender, S. (1974). Calculation of the wasserstein distance between probability distributions on the line. *Theory of Probability & Its Applications*, *18*(4), 784–786.

Wang, W., Dong, J., & Tan, T. (2009). Effective image splicing detection based on image chroma. *International Conference on Image Processing*.

Wu, R., Li, X., & Yang, B. (2011). Identifying computer generated graphics via histogram features. In *18th ieee international conference on image processing (icip)* (pp. 1933–1936).

Xu, J., Lei, B., Gu, Y., Winslett, M., Yu, G., & Zhang, Z. (2015). Efficient similarity join based on earth movers distance using mapreduce. *IEEE Transactions on Knowledge and Data Engineering*.

Zhou, L., Wang, D., Guo, Y., & Zhang, J. (2007). Blur detection of digital forgery using mathematical morphology. *Agent and Multi-Agent Systems: Technologies and Applications*.