FISH: Face Intensity-Shape Histogram Representation for Automatic Face Splicing Detection

Marco Fanfani[a,*], Fabio Bellavia[a], Massimo Iuliani[a,b], Alessandro Piva[a,b], Carlo Colombo[a]

[a] *Dept. of Information Engineering, University of Florence, Florence, Italy*
[b] *FORLAB Multimedia Forensics Laboratory, University of Florence, Prato, Italy*

Abstract

Tampered images spread nowadays over any visual media influencing our judgement in many aspects of our life. This is particularly critical for face splicing manipulations, where recognizable identities are put out of context. To contrast these activities on a large scale, automatic detectors are required.

In this paper, we present a novel method for automatic face splicing detection, based on computer vision, that exploits inconsistencies in the lighting environment estimated from different faces in the scene. Differently from previous approaches, we do not rely on an ideal mathematical model of the lighting environment. Instead, our solution, built upon the concept of histogram-based features, is able to statistically represent the current interaction of faces with light, untied from the actual and unknown reflectance model. Results show the effectiveness of our solution, that outperforms existing approaches on real-world images, being more robust to face shape inaccuracies.

Keywords: Image Forensics, Scene level analysis, Geometric Constraints, Lighting environment, Face splicing detection.

1. Introduction

Manipulated images are becoming ubiquitous in everyday life. Thanks to the advancement of photo-editing software, highly realistic tampering can be

*Corresponding author
Email address: marco.fanfani@unifi.it (Marco Fanfani)

Preprint submitted to Journal of Visual Communication and Image Representation July 19, 2019

produced even by non-expert users, with deep social impact and critical consequences in our perception of reality. In order to detect and contrast the spread of these fake images, Image Forensics has developed several solutions [1] aimed at determining if an image is pristine or tampered according to the presence, absence or inconsistency between the traces left by operations such as image acquisition, compression and other editing processes.

Face splicing, achieved by inserting into an original image a human face retrieved from a different photo, is one of the most critical tampering since it deals with people identity and can be used to produce images where specific subjects are inserted into an inconvenient and awkward context. *Signal level* traces found as invisible footprints into the signal statistics, such as demosaicing [2] or compression [3, 4] artifacts, or noise [5, 6], can be employed to detect face splicing. Unluckily, these solutions have a limited applicability, since the abovementioned traces may be partially or completely spoiled by common operations on images, such as resizing, compression, etc. [7]. More recently, alternative methods based on deep-learning [8] or exploiting the inconsistencies at the *physical level* of the scene represented in the image have arisen, considering shadows [9], perspective [10], or lighting [11, 12] incongruities.

In this paper we present a novel technique to detect face splicing based on physical-level analysis of the imaged scene. Previous works exploiting physical traces in the image try to directly extract and estimate the lighting parameters (i.e., the light source position, color and intensity) on each single face in the image, from which to detect inconsistencies indicating possible tampering. The major novelties of our approach are:

- Instead of a complex and partially incomplete ideal model characterizing the interaction of light with faces, we propose to employ histogram-based features. Histograms have proved to be very effective in many computer vision tasks [13] and, to the best of our knowledge, were never employed for face splicing detection;

- Novel ad-hoc metrics to compute distances between FISH features have also been designed, taking into account pixel saturation and albedo differences, so as to further improve the accuracy of our face splicing detector;

- Since our face features depend only on some image data statistics, without focusing on a particular mathematical model, on real images they outperform the state-of-the-art approach of [14] due to their higher robustness against image noise and face shape estimation inaccuracies;
- Finally, our approach is computationally more efficient, since it relies only on histogram computation, while the state-of-the-art requires complex face and lighting renderings.

The paper is organized as follows: In the next Section, a brief overview of the state-of-the-art methods is presented. The proposed histogram-based representation is described in Sect. 3, and used as the main building block for the fully automatic pipeline of Sect. 4. An experimental evaluation of our approach is reported in Sect. 5, and conclusions are finally drawn in Sect. 6.

2. State of the Art

Estimating the light source parameters of a real scene is quite a challenging task [15] which can prove extremely useful for detecting tampered images. In the recent literature on image forensics, some methods aim to detect image inconsistencies by estimating the color of the light source (i.e. the illuminant), while others focus on fitting a parametric model describing the interaction of the light source with the environment, for which the light source location/direction is usually the most relevant parameter.

The estimation of the light source color is strictly connected with the *colour constancy* problem [16], that requires to subtract the real light color from the input image in order to make the scene appear as it was acquired under a white illuminant. In the case of forensic applications, features related to light color are extracted on several patches of the images using the Gray-World assumption [17, 18], or physical-based solutions like the Inverse Intensity-Chromaticity [19] and compared across the image looking for anomalies. In particular, in [11] a SVM classifier is trained on features extracted from an illuminant map (i.e., a superpixel tessellation of the image, associating each patch to its illuminant color) computed by solving the color constancy problem.

Parametric models describing the interaction between light and the environment are based on the spherical harmonics representation [20, 21]. In particular, under the assumption of convex Lambertian surfaces with uniform albedo, linear camera response and distant light sources, for each color channel the light intensity $I(\mathbf{x}_k)$ measured at pixel $\mathbf{x}_k$ can be modeled as a linear combination of the spherical harmonics $\{Y_{n,m}(N(\mathbf{X}_k))\}$. Up to the second order, these are evaluated as

$$
\begin{aligned}
&Y_{0,0}(N(\mathbf{X}_k)) = \frac{1}{\sqrt{4\pi}} && Y_{1,-1}(N(\mathbf{X}_k)) = \sqrt{\frac{3}{4\pi}} y_k \\
&Y_{1,0}(N(\mathbf{X}_k)) = \sqrt{\frac{3}{4\pi}} z_k && Y_{1,1}(N(\mathbf{X}_k)) = \sqrt{\frac{3}{4\pi}} x_k \\
&Y_{2,-2}(N(\mathbf{X}_k)) = 3\sqrt{\frac{5}{12\pi}} x_k y_k && Y_{2,-1}(N(\mathbf{X}_k)) = 3\sqrt{\frac{5}{12\pi}} y_k z_k \\
&Y_{2,0}(N(\mathbf{X}_k)) = \frac{1}{2}\sqrt{\frac{5}{4\pi}} (3z_k^2 - 1) && Y_{2,1}(N(\mathbf{X}_k)) = 3\sqrt{\frac{5}{12\pi}} x_k z_k \\
&Y_{2,2}(N(\mathbf{X}_k)) = \frac{3}{2}\sqrt{\frac{5}{12\pi}} (x_k^2 - y_k^2) &&
\end{aligned}
\tag{1}
$$

In the above formulation, the pixel $\mathbf{x}_k$ is the projection of a surface 3D point $\mathbf{X}_k$, with normal $N(\mathbf{X}_k) = [x_k, y_k, z_k]$. The coefficients up to the second order of the spherical harmonics, i.e. $\ell_{n,m}$ with $n = \{0, 1, 2\}$ and $m = \{-n, n\}$, almost uniquely identify the lighting environment. In order to estimate them, the linear system $M\ell = \mathbf{I}$, or explicitly

$$\begin{bmatrix} r_{0,0}(N(\mathbf{X}_1)) & r_{1,-1}(N(\mathbf{X}_1)) & \dots & r_{2,2}(N(\mathbf{X}_1)) \\ r_{0,0}(N(\mathbf{X}_2)) & r_{1,-1}(N(\mathbf{X}_2)) & \dots & r_{2,2}(N(\mathbf{X}_2)) \\ \vdots & \vdots & \ddots & \vdots \\ r_{0,0}(N(\mathbf{X}_K)) & r_{1,-1}(N(\mathbf{X}_K)) & \dots & r_{2,2}(N(\mathbf{X}_K)) \end{bmatrix} \begin{bmatrix} \ell_{0,0} \\ \ell_{1,-1} \\ \vdots \\ \ell_{2,2} \end{bmatrix} = \begin{bmatrix} I(\mathbf{x}_1) \\ I(\mathbf{x}_2) \\ \vdots \\ I(\mathbf{x}_K) \end{bmatrix} \quad (2)$$

is solved, where $r_{0,0}(N(\mathbf{X}_k)) = \pi Y_{0,0}(N(\mathbf{X}_k))$, $r_{1,m}(N(\mathbf{X}_k)) = \frac{2\pi}{3} Y_{1,m}(N(\mathbf{X}_k))$, $r_{2,m}(N(\mathbf{X}_k)) = \frac{\pi}{4} Y_{2,m}(N(\mathbf{X}_k))$, and $K \geq 9$ pixel sampling locations $\mathbf{x}_k$ are used.

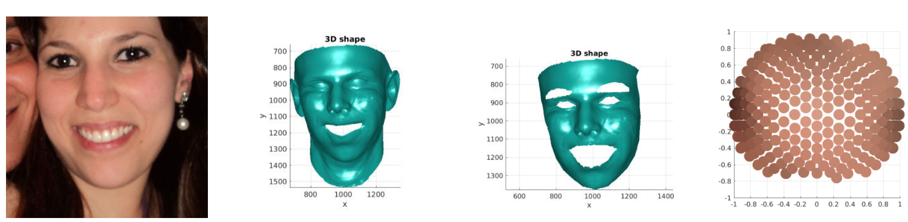
A possible splice is noticed when, in the same image, lighting coefficients relative to different parts of the scene exhibit relevant differences. In particular, lighting coefficients are estimated from occluding boundaries in [12], and from human faces in [22, 23, 14], after retrieving their 3D shape. To the best of our knowledge, the complex model described in [14], enriched to overcome the strict assumptions behind the spherical harmonics representation given above, is the current state-of-the-art in face splicing based on lighting observations. However, it still shows the main drawbacks inherent in retrieving the spherical lighting coefficients. More specifically, light estimation is very sensitive to the shape accuracy of the object upon which the matrix M is computed, i.e., the normals of the sampled points. This makes the solution very unstable, as can be noted by the performance degradation from synthetically rendered faces to real faces [14], for which the 3D shape is usually obtained automatically using morphable models [24, 25] or, more recently, deep learning [26]. Furthermore, still in the case of faces from real images, the advantages of using complex lighting models over simple ones are quite negligible.

According to these observations, and considering the difficulty in obtaining more accurate 3D models, in this paper we propose a different approach to face splicing based on an indirect estimation of the lighting map. In particular, instead of computing analytically the lighting coefficients, we build histograms relating surface normals with their intensity values, by statistically modelling the interaction map between light and the surface. The resulting descriptor design is inspired by histogram-based keypoint descriptors [13] employed in robust image matching. Indeed, the histograms associated to different faces are stable and robust to shape variations, and can be successfully used to indirectly measure lighting inconsistencies between spliced and pristine faces.

3. Face Intensity-Shape Histogram (FISH)

Under the assumption of convex and Lambertian surfaces with fixed albedo and distant light sources, the image intensity values of points in the scene only depend on their associated surface normals. In the case of faces, the resulting channel-wise mapping function $L : \mathbb{R}^3 \to \mathbb{R}$ from normals $\mathbf{n} = [x\, y\, z]^T$, $z > 0$ to a color channel intensity of the image $I = L(\mathbf{n})$ can be statistically modelled using a histogram-based representation, referred to as *Face Intensity-Shape Histogram* (FISH), computed as follows.

Given a face in the image and its associated 3D shape model (see Fig. 1a and 1b, respectively), we first pre-process the model so as to remove face regions strongly violating the assumptions above (see Fig. 1c). These regions include neck and ears (that yield poorly estimated normals), mouth, eyes and eyebrows (that have a different albedo and reflectance with respect to face skin), and saturated areas (i.e., pixels with maximum intensity among all channels out of the range $[15, 240]$ for 8-bit RGB images).



(a) (b) (c) (d)

Figure 1: (a) Detected face; (b) Registered 3D shape (using 3DMM); (c) Masked 3D shape; (d) FISH (best viewed in color).

FISH bins $i = 0, \ldots, \mathfrak{B}$ are sampled according to the vertices of a semiicosphere, that approximates a semi-sphere limited to the positive z-axis by a simplicial polyhedron at subdivision level 3 (i.e. an icosphere). Since an icosphere has 642 vertexes, of which only 305 with strictly positive z coordinate, it holds $\mathfrak{B} = 304$. Each bin corresponds to a distinct *quantized* surface normal $\mathbf{n}_i$ (see Fig. 1d). FISH bin values $I_i = L(\mathbf{n}_i)$ for each color channel are computed via Gaussian kernel density estimation as explained hereafter. Let $\hat{\mathbf{n}}_k = N(\mathbf{X}_k)$ and $\hat{I}_k = I(\mathbf{x}_k)$ be respectively the 3D shape normal vector of $\mathbf{X}_k$ and the intensity value of a pixel $\mathbf{x}_k$, which is the projection of $\mathbf{X}_k$ as in Eq. 2. (Notice that index i refer to bins, while index k to pixels/normals sampled on the face.) Then

$$I_i = \sum_k \frac{w_{ik}}{w_i} \hat{I}_k \tag{3}$$

where the sum is over the masked face pixels, with weights

$$w_i = \sum_k w_{ik} \tag{4}$$

computed from the Gaussian distribution

$$z_{ik} = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{\arccos(\mathbf{n}_i \cdot \hat{\mathbf{n}}_k)}{2\sigma}\right)^2} \tag{5}$$

subject to a influence cutoff threshold τ_k:

$$w_{ik} = \begin{cases} z_{ik} & \text{if } z_{ik} > \tau_k \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

The value of τ_k corresponds to the 2.5^{th} percentile of the distribution of the z_{ik}, for $i = [0, \ldots, \mathfrak{B}]$. In this way, weights associated to normals $\hat{\mathbf{n}}_k$ that are too far from the i^{th} bin representative $\mathbf{n}_i$ are forced to zero. The standard deviation σ used to define the kernel bandwidth in Eq. 5 is equal to $3/8$ times the average angular distance between two adjacent vertexes of the icosphere.

By concatenating the bin values for each channel, i.e.,

$$\boldsymbol{I} = \boldsymbol{L}(\mathbf{n}_i) = [L_{\mathrm{R}}(\mathbf{n}_i)\ L_{\mathrm{G}}(\mathbf{n}_i)\ L_{\mathrm{B}}(\mathbf{n}_i)]^T \tag{7}$$

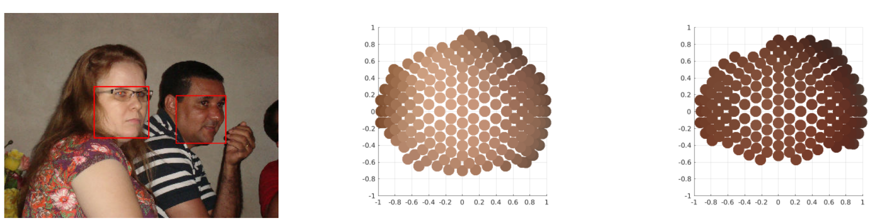
the final FISH descriptor $\boldsymbol{L}$ is obtained.

FISH descriptors can be used to compare faces in a probe image. The more two FISH descriptors are similar, the more the corresponding faces are likely to be exposed to the same lighting conditions. A possible definition of the distance $\mathcal{D}(a, b)$ between two FISH descriptors $\boldsymbol{L}^a$ and $\boldsymbol{L}^b$ associated to faces a and b is

$$\mathcal{D}(a, b) = \left(\sum_{\substack{i=0,\ldots\mathfrak{B} \\ (w_i^a>0)\wedge(w_i^b>0)}} \left\| \boldsymbol{I}_i^a - \boldsymbol{I}_i^b \right\|^2 \right)^{\frac{1}{2}} \tag{8}$$

where $\boldsymbol{I}_i^a = \boldsymbol{L}^a(\mathbf{n}_i)$, $\boldsymbol{I}_i^b = \boldsymbol{L}^b(\mathbf{n}_i)$, $\|\cdot\|$ is the Euclidean norm L_2—chosen experimentally, as it gives the best results among L_1, L_2, Wave edges, Canberra, Correlation, Bhattacharyya and Kullback Leibler—and w_i^a, w_i^b are defined as in Eq. 4. Notice that the above definition of $\mathcal{D}(a, b)$ takes explicitly into account the presence of empty histogram bins.

As shown in Fig. 2, unhandled skin albedo would result in an incorrect FISH-based face matching.



(a) (b) (c)

Figure 2: (b),(c): In the absence of skin tone normalization, the FISH descriptors for two faces in a pristine image (a) look different to each other, while they should not. (Best viewed in color.)

In order to remove skin color effects when comparing two FISH descriptors $\boldsymbol{L}^a$ and $\boldsymbol{L}^b$, we developed and tested two normalization strategies. The first strategy consists of simply pre-normalizing $\boldsymbol{L}$ by the mean RGB value $\boldsymbol{\mu}$ of the associated masked face, under the common assumption that albedo is a scale factor, i.e.,

$$\mathring{I}_i = \mathring{\boldsymbol{L}}(\mathbf{n}_i) = \boldsymbol{L}(\mathbf{n}_i)/\mu \tag{9}$$

channel-wise, so that

$$\mathcal{D}'(a,b) = \mathcal{D}(\mathring{\boldsymbol{L}}^a, \mathring{\boldsymbol{L}}^b) \quad . \tag{10}$$

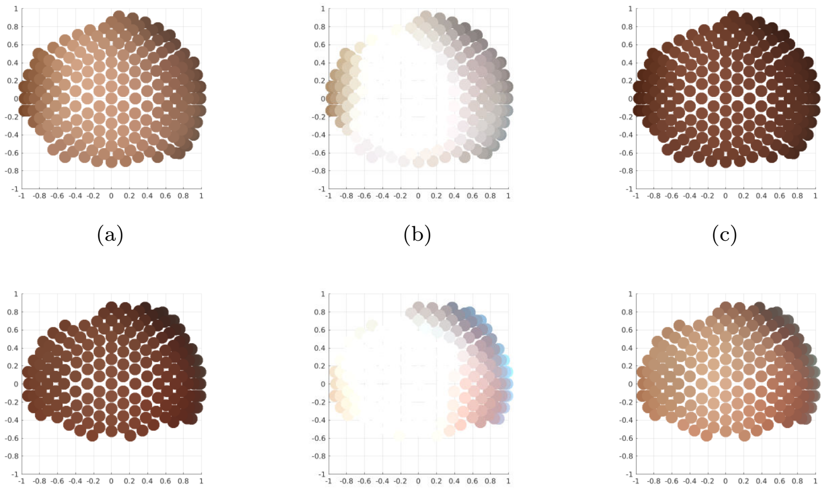
In the second strategy, color saturation is taken into account. In detail, the FISH descriptor $\boldsymbol{L}^a$ is normalized with respect to its albedo μ_a, then the albedo μ_b of $\boldsymbol{L}^b$ is applied, clipping saturated values, i.e.,

$$I_i^{a\to b} = \boldsymbol{L}^{a\to b}(\mathbf{n}_i) = \min(255, \boldsymbol{L}^a(\mathbf{n}_i)\,\frac{\mu_b}{\mu_a}) \tag{11}$$

$\boldsymbol{L}^{a\to b}$ is then compared with $\boldsymbol{L}^b$. The final distance is made symmetric by also considering the case in which the μ_a is applied to $\boldsymbol{L}^b$, so that

$$\mathcal{D}''(a,b) = \min(\mathcal{D}(\boldsymbol{L}^a, \boldsymbol{L}^{b\to a}), \mathcal{D}(\boldsymbol{L}^b, \boldsymbol{L}^{a\to b})) \tag{12}$$

Referring to Fig. 3 we present an example of both normalization strategies for the faces of the pristine image in Fig. 2. Fig. 3a and 3d show the FISHs $\boldsymbol{L}^a$ and $\boldsymbol{L}^b$ without any normalization: Their comparison produces a distance of 65.42. In this case, the effect of the skin color strongly affects the distance, introducing a bias related to the face albedo. This can be suppressed by normalizing each descriptor with its mean RGB value, thus obtaining the FISHs $\mathring{\boldsymbol{L}}^a$ and $\mathring{\boldsymbol{L}}^b$, shown in Fig. 3b and 3e. Comparing these normalized descriptors yields a distance of 25.07. However, $\mathring{\boldsymbol{L}}^a$ and $\mathring{\boldsymbol{L}}^b$ cannot take into account saturated values that go outside the range $[0, 255]$. In this case, using the FISH descriptors $\boldsymbol{L}^{a\to b}$ (Fig. 3c) and $\boldsymbol{L}^{b\to a}$ (Fig. 3f) can handle this saturation side-effects. In particular, to compute $\mathcal{D}''(a,b)$, we first evaluate the distance between $\boldsymbol{L}^b$ and $\boldsymbol{L}^{a\to b}$ (i.e. Fig. 3d and 3c), and between $\boldsymbol{L}^a$ and $\boldsymbol{L}^{b\to a}$ (i.e. Fig. 3a and 3f), and



(a) (b) (c)

(d) (e) (f)

Figure 3: Normalized descriptor obtained from the pristine image of Fig. 2. While the distance $\mathcal{D}$ without any normalizations ((a) and (d)) obtains a score of 65.42, $\mathcal{D}'$ ((b) and (e)) lowers the score to 25.07. Finally, $\mathcal{D}''$ obtains 7.02 as the minimum between 13.45 (from (a) and (f)) and 7.02 (from (c) and (d)). (Best viewed in color.)

then we select the minimum among the two distances, that in this case is 7.02.

Figure 4 shows an example face, together with results synthesized from the inverse mapping of the FISH model and from the spherical harmonics coefficients obtained as described in [27]. Since the FISH model preserves better shading details than the spherical harmonics model, FISH fits better real data, which also implies an implicit relaxation of the strict assumptions defining the interaction of light with the environment.

(a) (b) (c) (d)

Figure 4: Examples of inverse synthesized face. (a) Original image; (b) Masked face; (c) FISH reverse mapping synthesis; (d) spherical harmonics synthesis. (Best viewed in color.)

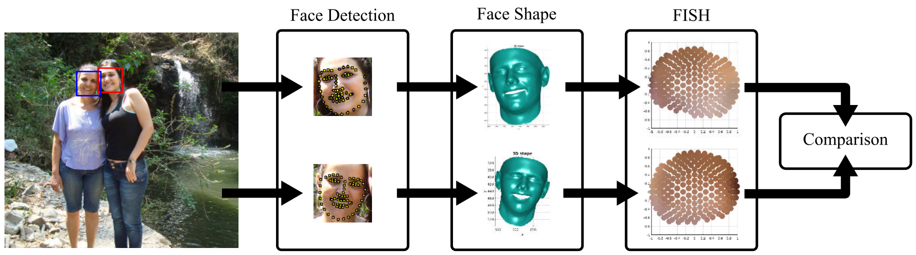


Figure 5: Pipeline for automatic face splicing detection using FISH descriptors.

4. Automatic face splicing detection pipeline

We employed the FISH descriptor to develop a fully automated pipeline for face splicing detection, that can be divided into the following three steps (see Fig. 5):

- **Face detection**. The method proposed in [28] is used, which exploits general Deformable Part Models trained to specifically detect faces. Subparts of the object are detected by taking into account the deformation with respect to a mean shape (detection threshold is set to 0.3). From each detected face region, 68 face landmarks are successively localized according to the face alignment algorithm of [29], based on Supervised Descent Method, used with the default parameters.

- **Face shape and normals estimation**. Face landmarks computed at the previous step are used to register a *3D Morphable Model* (3DMM) and to obtain an estimate of the face shape. In particular, we adopted the solution presented in [30], combining the *Basel Face Model* [24] and the *Face Warehouse* model [25] in order to be able to adapt the model to both identity and expression. As an alternative approach, we also tested the recent method proposed in [26] based on convolutional neural networks.

- **FISH descriptors extraction and comparison**. See Sect. 3.

Note that, since our method, as well as [14] and [27], compares lighting estimates to detect discrepancies, at least two faces are required. Moreover, in the case that only two faces are detected, the pipeline can detect the occurrence of tampering, but is unable to indicate which of the two is the tampered face, while, if more than two faces are found, the spliced face can be localized as the one with the greatest distance in terms of FISH descriptors from the other faces. Notice also that it is assumed that all the subjects under analysis are subjected to the same lighting environment.

5. Experimental evaluation

In order to gain a deep insight into FISH performance, several comparative tests were carried out using different datasets that cover increasing levels of complexity, from a fully synthetic setup (Sect. 5.1), through a controlled face acquisition setup with manual 3D model estimation (Sect. 5.2), to a real-world, unconstrained scenario (Sect. 5.3).

5.1. Synthetically generated faces

This evaluation employs the Syn1 and Syn2 datasets, presented in [14], where two sets of 3D synthetic faces have been rendered with known random lights. Since FISH does not compute spherical harmonics, a direct estimation of the error in terms of lighting coefficients as in [14] cannot be done. Nevertheless, a higher distance between the related FISH descriptors must be expected as the discrepancy in two lighting environments increases. Under this observation, the correlation between the difference of two ground-truth spherical harmonics vectors, corresponding to the two faces to be checked, and the distance of the related FISH descriptors, provides a good indicator of the method accuracy. For this scope, we created *virtually* spliced probes by considering two faces with different lighting, and evaluated the correlation between the scores obtained by FISH and the ground-truth values in terms of Spearman's rank correlation coefficient (SROCC). Additionally, in order to evaluate the method robustness w.r.t. noise in the images and in the 3D shape estimates, the evaluation was repeated by injecting Gaussian noise with zero mean and variable standard deviation σ. In particular, a Gaussian noise with $\sigma_{\mathrm{RGB}} = \{5, 7\}$ was added to each RGB channel independently, and similarly a Gaussian noise with $\sigma_{\mathrm{N}} = \{0.1, 0.2, 0.3, 0.4, 0.5\}$ was added to each normal vector dimension independently.

Table 1 reports the results obtained by FISH and the baseline method of [27]. For our pipeline using FISH descriptors, the superscript '†' (i.e. FISH†) indicates that no mask is applied to the saturated pixels.

Table 1: SROCC on Syn1 Syn2 (best results in bold)

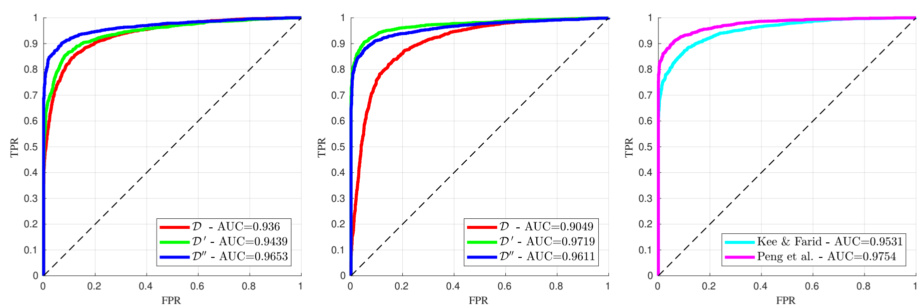
Method	**Original**	**Image noise**		**Shape noise**				
		$\sigma_{\mathrm{RGB}} = 5$	$\sigma_{\mathrm{RGB}} = 7$	$\sigma_{\mathrm{N}} = 0.1$	$\sigma_{\mathrm{N}} = 0.2$	$\sigma_{\mathrm{N}} = 0.3$	$\sigma_{\mathrm{N}} = 0.4$	$\sigma_{\mathrm{N}} = 0.5$
FISH† with $\mathcal{D}$	0.7639	0.7639	0.7636	0.7670	0.7492	0.7170	0.6738	0.6191
FISH† with $\mathcal{D}'$	0.8625	0.8626	0.8620	0.8608	0.8457	**0.8278**	0.8057	**0.7941**
FISH† with $\mathcal{D}''$	0.8544	0.8545	0.8538	0.8484	0.8288	0.8077	0.7846	0.7673
FISH with $\mathcal{D}$	0.7639	0.7639	0.7636	0.7671	0.7491	0.7170	0.6738	0.6192
FISH with $\mathcal{D}'$	**0.8627**	**0.8628**	**0.8621**	**0.8609**	**0.8459**	**0.8278**	**0.8059**	0.7940
FISH with $\mathcal{D}''$	0.8543	0.8545	0.8539	0.8485	0.8289	0.8078	0.7846	0.7672
Kee & Farid [27]	0.8131	0.8135	0.8137	0.8183	0.8127	0.7896	0.7557	0.7365

As shown in the table, FISH correlation with light coefficients is high, in particular using the distance normalization schemes $\mathcal{D}'$, and $\mathcal{D}''$. FISH with distance normalizations has better correlation than the baseline spherical harmonics estimation method of [27] also when noise is added. Note that FISH and FISH† obtain very close results, since for these images no saturated pixels are present (i.e. there are not highlights or strong shadows). Results with the method of [14] are not reported in Table 1 since nothing can actually be said about the behavior of this approach in the presence of noise. Indeed, this method does not use the normal vectors directly: It requires to render the face 3D model on 42 images with different lightings and estimate the optimized transfer coefficients. This can only be done with the knowledge of additional data, unavailable to us. If no noise is present, the solution of [14] obtains a very high correlation value (0.9592), thanks to the availability of the original true 3D face model for the rendering process, which actually is an unrealistic scenario in practical situations.

5.2. Real faces in a controlled acquisition setup

For this test, the Yale Face Database B (YaleB) [31] was used, that includes a set of images obtained from 10 distinct faces captured in different poses under

49 different lighting conditions. Following [14], we focused on frontal faces, thus reducing the dataset to 490 test images. Analogously to the previous experimental evaluation on Syn1 and Syn2, a *virtually* spliced dataset was generated by considering for the negative (pristine) set all the face pairs of different identities with the same lighting, obtaining $(49 \times 10 \times 9)/2 = 2205$ pristine images. On the other hand, there are $(49 \times 10 \times 48 \times 9)/2 = 105840$ tampered probes, from which the positive (spliced) set was generated by randomly sampling a number of examples equal to that of the negative class. (A similar experiment was carried out in [14], where the authors randomly sampled 10000 probes for both the negative and positive classes, thus introducing repetitions in the negative class. Hence the slight discrepancies between our results and theirs.)



(a) (b) (c)

Figure 6: ROC curves for the *virtual* splicing test on YaleB: (a) FISH†, (b) FISH, where the three distance $\mathcal{D}$, $\mathcal{D}'$, $\mathcal{D}''$ and reported respectively in red, green and blue. In (c) ROCs for [14] and [27]. (best viewed in color)

Figure 6 reports the Receiver Operating Characteristic (ROC) plots for our FISH and FISH†, using all the distances $\mathcal{D}$, $\mathcal{D}'$, and $\mathcal{D}''$, together with results from [14] and [27], obtained by using the code available online. The Area Under the Curve (AUC) is reported in Table 2 for completeness, together with the True Positive Rate (TPR) at 0.01, 0.05, and 0.10 False Positive Rate (FPR). For this controlled acquisition setup on real face images, all the methods obtained comparable results. Notice that for this test, high-quality 3D face shapes were computed using Face Gen[1], which requires several input images from different views for face, and manually annotated landmarks. It is worth remarking that this is still an unrealistic application scenario for us, as we target to work with real and noisy images on an automatic pipeline.

Table 2: Tests on YaleB (best results in bold)

Method	AUCs	TPR @ 0.01 FPR	TPR @ 0.05 FPR	TPR @ 0.10 FPR
FISH† with $\mathcal{D}$	0.9360	0.5066	0.7315	0.8295
FISH† with $\mathcal{D}'$	0.9439	0.6390	0.7864	0.8671
FISH† with $\mathcal{D}''$	0.9653	0.7950	0.8739	0.9161
FISH with $\mathcal{D}$	0.9049	0.1887	0.5633	0.7592
FISH with $\mathcal{D}'$	0.9719	0.8127	**0.9034**	**0.9356**
FISH with $\mathcal{D}''$	0.9611	0.7923	0.8739	0.9120
Peng et al. [14]	**0.9754**	**0.8345**	0.8961	0.9311
Kee & Farid [27]	0.9531	0.7120	0.8082	0.8680

5.3. Real faces in the wild

Tests with a fully unconstrained scenario were carried out by evaluating our automated pipeline on the DSO-1 dataset [11] containing real images. The DSO-1 dataset includes 100 pristine and 100 spliced images, with challenging manipulations. The dataset shows high variation of people poses and expressions, captured in indoor and outdoor scenarios under uncontrolled lighting conditions. Occlusions caused by other faces or objects (like glasses or hair) are also present. To the best of our knowledge, DSO-1 is the only freely available real-world face splicing database.

In order to compare our results with those reported in [14], we strictly followed their protocol[2], by excluding some DSO-1 images and by limiting the comparison to face pairs.

Table 3 reports the AUC of the ROC curve for different versions of our method and the current state-of-the-art methods. For our pipeline using FISH descriptors, the superscript '$\star$' is applied when the recent CCN method described in [26] is employed to compute the 3D face model instead of the standard 3DMM. Figure 7 also reports ROC curves for our pipelines.

[1] `https://facegen.com/modeller.htm`
[2] `https://github.com/bomb2peng/CASIA_3Dlighting/tree/master/datasets/DSO-1`

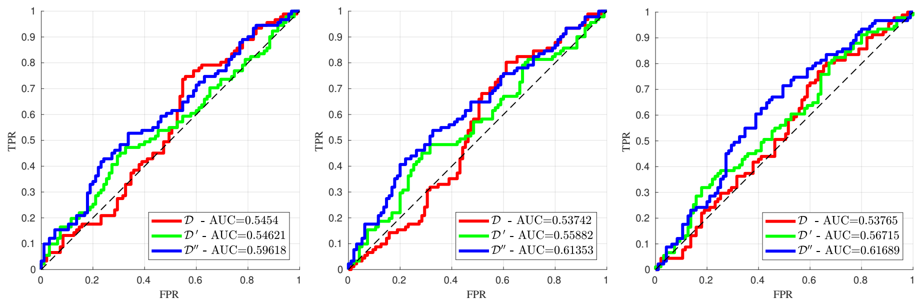
Table 3: Face splicing detection in terms of AUC on the DSO-1 dataset (best results in bold). Results for the state-of-the-art methods have been retrieved from [14]

Method	**AUC**
FISH† with $\mathcal{D}$	0.5454
FISH† with $\mathcal{D}'$	0.5462
FISH† with $\mathcal{D}''$	**0.5962**
FISH with $\mathcal{D}$	0.5374
FISH with $\mathcal{D}'$	0.5588
FISH with $\mathcal{D}''$	**0.6135**
FISH* with $\mathcal{D}$	0.5376
FISH* with $\mathcal{D}'$	0.5672
FISH* with $\mathcal{D}''$	**0.6169**
Peng et al. [14]	0.5795
Kee & Farid [27]	0.5715
Fan et al. [32]	0.5633

Results show that all the methods based on FISH obtain a better AUC with respect to the state-of-the-art in combination with the $\mathcal{D}''$ distance, demonstrating the effectiveness of the proposed solution. Exclusion of saturated pixels produce an additional improvement, while the albedo handling mechanism is very critical, as shown by the changes of performance when employing $\mathcal{D}$, $\mathcal{D}'$ and $\mathcal{D}''$. Moreover, while FISH* does not considerably improve the results with respect to the other FISH variants, as it lowers the False Positive Rate (FPR) but also slightly decreases the True Positive Rate (TPR), nevertheless it benefits from a minor computational complexity and code management over FISH. In addition, the FISH descriptor can better handle errors on the 3D shape clustering and in weighting the contributions of similar normal vectors, thus reducing the impact of incorrectly estimated normals. For this reason, FISH can be more reliable in a fully automatic scenario, where the accuracy of the 3D face model is lower than in a synthetic scenario.

5.4. Distance normalization on FISH

As it can be noticed from experiments reported in Sec. 5.1 and Sec. 5.2, in all the tests on the Syn1 and Syn2 and using FISH† on the Yale database, the best results are achieved with the $\mathcal{D}'$ distance, while using FISH on Yale and



(a) (b) (c)

Figure 7: ROC curves on DSO-1 with respectively (a) FISH†, (b) FISH, and (c) FISH*. For each version, the three distance $\mathcal{D}$, $\mathcal{D}'$, $\mathcal{D}''$ and reported respectively in red, green and blue (best viewed in color).

in all cases on the DSO-1 dataset, it is $\mathcal{D}''$ that obtains the best scores. This behavior is reasonably due to the different ranges of RGB values that can be found in the images. Table 4 reports for each dataset the standard deviation of the average RGB color of the related faces with and without saturated values. The standard deviation values are computed over the mean RGB value of each face, considering all the pixels exploited to compute the FISH descriptor (i.e. all pixels that are projection of a 3D vertex of the face model).

Table 4: Standard deviation of the average RGB color of the faces. Note that for YaleB only gray-scale images are provided

With saturated pixels

Dataset	STD		
	R	G	B
Syn1	12.99	9.91	8.92
Syn2	12.99	9.91	8.92
YaleB	23.93		
DSO-1	29.99	29.54	28.58

Without saturated pixels

Dataset	STD		
	R	G	B
Syn1	12.99	9.91	8.92
Syn2	12.99	9.91	8.92
YaleB	16.02		
DSO-1	25.21	23.79	23.13

According to the table, $\mathcal{D}'$ gives better results in the case of low variance (e.g. inferior to 20), while $\mathcal{D}''$ obtains better results for data with higher variance. Notice also that no saturated pixels are found in the synthetic datasets, which confirms their limits in simulating a real scenario.

5.5. Computational complexity

Both FISH and the methods of [14] and [27] share the initial steps of the pipeline (i.e., face detection and alignment, and 3D shape estimation). These steps take most of the time spent in computation, that in our Matlab implementation correspond respectively on about 9 seconds for face detection on each image, plus 0.15 and 0.08 seconds for face alignment and 3DMM fitting for each single face detected.

Additionally, FISH and [27] just require to estimate the normal vectors of the face shape, which takes about 10 seconds on average on our Matlab non optimized implementation, while [14] exploits 3D information to synthesize 42 images of the face under different known illuminations in order to estimate the *transfer coefficients* that are exploited to retrieve the lighting vector. Although we cannot effectively verify the computational time spent by [14] as we lack data to replicate this step, it would reasonably be equal or surpass the time spent by FISH, since rendering software typically has to estimate the shape normal vectors in addition to other steps. Moreover, [14] also requires to solve N 42x9 linear systems (i.e., 42 images per 9 lighting transfer functions, for each of the sampling points).

For the final step, both methods in [27] and [14] solve a linear system with N equations, that in our implementation takes about 5 milliseconds. On the other hand, the FISH histogram has a computational complexity of $O(N\mathfrak{B})$, that in our non-optimized implementation takes about 80 milliseconds.

Considering the whole pipeline, FISH running times are comparable to those of [27], since most of the time is spent in the first step of the pipeline, while [14] should spend more time for the computation of the *transfer coefficients*.

Notice that the distance computation is slightly slower for our solution, due to the higher dimension of the histogram w.r.t. the lighting vector, but this has a negligible impact over the computation time for the whole pipeline.

5.6. Limitations

FISH splicing detection, similarly to [14] and [27], relies on the comparison of physical lighting environments from distinct faces, and requires at least two faces in a probe image. Additionally, this kind of approach would not work if the scene strongly violates the assumption of Lambertian surfaces illuminated by distant lights, such in the case when objects in the scene cast strong shadows over one of the faces under inspection. Finally, image resolution should be sufficiently high to allow accurate face alignment and sampling of light color intensity data.

6. Conclusion

This paper presented a novel approach to face splicing detection based on light analysis. The proposed FISH descriptor is designed according to a statistical representation based on histograms, implicitly estimating the mapping between image intensities and 3D normal vectors. FISH can alleviate the impact of the low accuracy of the 3D face model, which typically strongly affects the methods based on spherical harmonics. The effectiveness and robustness of our solution has been demonstrated on three different datasets: While in the controlled scenarios of Syn1/Syn2 and YaleB FISH obtains results comparable to the state-of-the-art, on images acquired on real scenarios with unconstrained lighting conditions, such those of the DSO-1 dataset, it outperform all the existing face splicing detectors based on lighting analysis.

Acknowledgments

This material is based on research 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.

References

[1] A. Piva, An overview on image forensics, ISRN Signal Processing 2013 (2013) Article ID 496701, 22 pages.

[2] P. Ferrara, T. Bianchi, A. De Rosa, A. Piva, Image forgery localization via fine-grained analysis of cfa artifacts, Information Forensics and Security, IEEE Transactions on 7 (5) (2012) 1566–1577. doi:10.1109/TIFS.2012.2202227.

[3] B. Li, T. Ng, X. Li, S. Tan, J. Huang, Revealing the trace of high-quality JPEG compression through quantization noise analysis, Information Forensics and Security, IEEE Transactions on 10 (3) (2015) 558–573. doi:10.1109/TIFS.2015.2389148.

[4] T. Bianchi, A. Piva, Image forgery localization via block-grained analysis of jpeg artifacts, Information Forensics and Security, IEEE Transactions on 7 (3) (2012) 1003–1017. doi:10.1109/TIFS.2012.2187516.

[5] M. Chen, J. Fridrich, M. Goljan, J. Lukas, Determining image origin and integrity using sensor noise, Information Forensics and Security, IEEE Transactions on 3 (1) (2008) 74–90. doi:10.1109/TIFS.2007.916285.

[6] B. Mahdian, S. Saic, Using noise inconsistencies for blind image forensics, Image and Vision Computing 27 (10) (2009) 1497 – 1503.

[7] M. Zampoglou, S. Papadopoulos, Y. Kompatsiaris, Detecting image splicing in the wild (web), in: Proc. IEEE Int Multimedia & Expo Workshops (ICMEW) Conf, 2015, pp. 1–6.

[8] P. Zhou, X. Han, V. I. Morariu, L. S. Davis, Two-stream neural networks for tampered face detection, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2017, pp. 1831–1839.

[9] E. Kee, J. F. O'Brien, H. Farid, Exposing photo manipulation with inconsistent shadows, ACM Trans. Graph. 32 (3) (2013) 28:1–28:12. doi:10.1145/2487228.2487236.

[10] M. Iuliani, M. Fanfani, C. Colombo, A. Piva, Reliability assessment of principal point estimates for forensic applications, J. Visual Communication and Image Representation 42 (2017) 65–77.

[11] T. Carvalho, C. Riess, E. Angelopoulou, H. Pedrini, A. de Rezende Rocha, Exposing digital image forgeries by illumination color classification, IEEE Transactions on Information Forensics and Security (2013) 1182–1194.

[12] M. Johnson, H. Farid, Exposing digital forgeries in complex lighting environments, Information Forensics and Security, IEEE Transactions on 2 (3) (2007) 450 –461. doi:10.1109/TIFS.2007.903848.

[13] F. Bellavia, C. Colombo, Rethinking the sGLOH descriptor, IEEE Transactions on Pattern Analysis and Machine Intelligence 40 (4) (2018) 931–944.

[14] B. Peng, W. Wang, J. Dong, T. Tan, Optimized 3d lighting environment estimation for image forgery detection, IEEE Transactions on Information Forensics and Security 12 (2) (2017) 479–494. doi:10.1109/TIFS.2016.2623589.

[15] W. Zhou, C. Kambhamettu, A unified framework for scene illuminant estimation, Image and Vision Computing 26 (3) (2008) 415 – 429.

[16] A. Gijsenij, T. Gevers, J. van de Weijer, Computational color constancy: Survey and experiments, IEEE Transactions on Image Processing 20 (9) (2011) 2475–2489. doi:10.1109/TIP.2011.2118224.

[17] G. Buchsbaum, A spatial processor model for object colour perception, Journal of the Franklin Institute 310 (1) (1980) 1 – 26. doi:https://doi.org/10.1016/0016-0032(80)90058-7.
URL http://www.sciencedirect.com/science/article/pii/0016003280900587

[18] J. van de Weijer, T. Gevers, A. Gijsenij, Edge-based color constancy, IEEE Transactions on Image Processing 16 (9) (2007) 2207–2214. doi:10.1109/TIP.2007.901808.

[19] R. T. Tan, K. Nishino, K. Ikeuchi, Color constancy through inverse-intensity chromaticity space, J. Opt. Soc. Am. A 21 (3) (2004) 321–334. doi:10.1364/JOSAA.21.000321.

[20] R. Ramamoorthi, P. Hanrahan, On the relationship between radiance and irradiance: determining the illumination from images of a convex lambertian object, J. Opt. Soc. Am. A 18 (10) (2001) 2448–2459. doi:10.1364/JOSAA.18.002448.
URL http://josaa.osa.org/abstract.cfm?URI=josaa-18-10-2448

[21] R. Basri, D. W. Jacobs, Lambertian reflectance and linear subspaces, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (2) (2003) 218–233. doi:10.1109/TPAMI.2003.1177153.

[22] B. Peng, W. Wang, J. Dong, T. Tan, Improved 3d lighting environment estimation for image forgery detection, in: 2015 IEEE International Workshop on Information Forensics and Security (WIFS), 2015, pp. 1–6. doi:10.1109/WIFS.2015.7368587.

[23] B. Peng, W. Wang, J. Dong, T. Tan, Automatic detection of 3d lighting inconsistencies via a facial landmark based morphable model, in: 2016 IEEE International Conference on Image Processing (ICIP), 2016, pp. 3932–3936.

[24] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, T. Vetter, A 3d face model for pose and illumination invariant face recognition, in: 2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance, 2009, pp. 296–301.

[25] C. Cao, Y. Weng, S. Zhou, Y. Tong, K. Zhou, Facewarehouse: A 3d facial expression database for visual computing, IEEE Transactions on Visualization and Computer Graphics 20 (3) (2014) 413–425. doi:10.1109/TVCG.2013.249.
URL http://dx.doi.org/10.1109/TVCG.2013.249

[26] G. Trigeorgis, P. Snape, I. Kokkinos, S. Zafeiriou, Face normals in-the-wild using fully convolutional networks, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 340–349. doi:10.1109/CVPR.2017.44.

[27] E. Kee, H. Farid, Exposing digital forgeries from 3-d lighting environments, in: 2010 IEEE International Workshop on Information Forensics and Security, 2010, pp. 1–6. doi:10.1109/WIFS.2010.5711437.

[28] M. Mathias, R. Benenson, M. Pedersoli, L. Van Gool, Face detection without bells and whistles, in: Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV, Springer International Publishing, Cham, 2014, pp. 720–735. doi:10.1007/978-3-319-10593-2_47.
URL https://doi.org/10.1007/978-3-319-10593-2_47

[29] X. Xiong, F. D. la Torre, Supervised descent method and its applications to face alignment, in: 2013 IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 532–539. doi:10.1109/CVPR.2013.75.

[30] X. Zhu, Z. Lei, J. Yan, D. Yi, S. Z. Li, High-fidelity pose and expression normalization for face recognition in the wild, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 787–796. doi:10.1109/CVPR.2015.7298679.

[31] A. S. Georghiades, P. N. Belhumeur, D. J. Kriegman, From few to many: illumination cone models for face recognition under variable lighting and pose, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (6) (2001) 643–660.

[32] W. Fan, K. Wang, F. Cayre, Z. Xiong, 3d lighting-based image forgery detection using shape-from-shading, in: 2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO), 2012, pp. 1777–1781.