

# A Robust 3D-based Color Correction Approach for Texture Mapping Applications

Daniel Coelho <sup>1,2\*</sup> , Lucas Dal'Col <sup>1,2</sup> , Tiago Madeira <sup>2</sup> , Paulo Dias <sup>2,3</sup>  and Miguel Oliveira <sup>1,2</sup> 

<sup>1</sup> Department of Mechanical Engineering, University of Aveiro, 2810-193 Aveiro, Portugal; danielsilveiracoelho@ua.pt (D.C.); lucasrdalcol@ua.pt (L.D.); mriem@ua.pt (M.O)

<sup>2</sup> Institute of Electronics and Informatics Engineering of Aveiro, University of Aveiro, 3810-193 Aveiro; tiagomadeira@ua.pt (T.M), paulo.dias@ua.pt (P.D)

<sup>3</sup> Department of Electronics, Telecommunications and Informatics, University of Aveiro, 3810-193 Aveiro, Portugal

\* Correspondence: danielsilveiracoelho@ua.pt

**Abstract:** Texture mapping of 3D models using multiple images often results in textured meshes with unappealing visual artifacts known as texture seams. These artifacts can be more or less visible, depending on the color similarity between the used images. The main goal of this work is to produce textured meshes free of texture seams through a process of color correcting all images of the scene. To accomplish this goal, we propose two contributions to the state of the art of color correction: a pairwise-based methodology, capable of color correcting multiple images from the same scene; the application of 3D information from the scene, namely meshes and point clouds, to build a filtering procedure, in order to produce a more reliable spatial registration between images, thereby increasing the robustness of the color correction procedure. We also present a texture mapping pipeline that receives uncorrected images, an untextured mesh, and point clouds as inputs, producing a final textured mesh and color corrected images as output. Results include a comparison with four other color correction approaches. These show that the proposed approach outperforms all others, both in qualitative and quantitative metrics. The proposed approach enhances the visual quality of textured meshes by eliminating most of the texture seams.

**Keywords:** color correction, texture mapping, joint image histogram, color mapping function

## 1. Introduction

The creation of 3D models from the captured shape and appearance of objects is known as 3D reconstruction. It is a widely researched topic in areas such as computer graphics [1] and computer vision [2], and has recently gained attention in others, namely robotics [3], autonomous driving [4], medical applications [5], cultural heritage [6] and agriculture [7].

Texture mapping is the colorization of a 3D mesh using a single image. However, in more recent applications, texture mapping has been applied to cases where several overlapping images are available to colorize the mesh. The overlap will often occur in irregular regions, which creates an intricate combination of dependencies. This entails the existence of a mechanism to handle the redundant photometric information, and is often referred to as multi-view texture mapping [8–11]. The problem is generally solved in two different ways: by selecting a single image from the set of possible images, or by fusing the texture from all available images. Several authors have proposed methodologies to carry out the fusion, based on different forms of weighted average of the contributions of textures in the image space [12,13]. However, these approaches are highly sensitive to inaccuracies in camera pose estimation, as even slight misalignments may generate ghost and blurring artefacts in the textures, which are not visually appealing [14]. Still, selecting a single image to be used for texture mapping raises the problem of choosing the most adequate one from a discrete set of possibilities and additionally, how to produce a consistent selection

**Citation:** Coelho, D.; Dal'Col, L.; Madeira, T.; Dias, P.; Oliveira, M. Title. *Sensors* **2022**, *1*, 0. <https://doi.org/>

Received:

Accepted:

Published:

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Copyright:** © 2022 by the authors. Submitted to *Sensors* for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

of images for all the faces in the 3D mesh. Some authors propose to employ a Markov random field to select the images [15]. Others have used graph optimization mechanisms [16], or optimization procedures that minimize discontinuities between neighbouring faces [17]. These approaches address the problem of blurring and ghost artefacts. However, particularly in regions where the borders of the selected images meet, visual seam artefacts are often noticeable. The seams are more or less visible depending on the similarity of colors in the images, which is why a proper color correction of the images is crucial to achieve seamless texture mapping. This may be carried out using a form of post-processing operation. For example, in [18] authors propose to use Poisson image editing techniques to smooth the texture. Color correction can be defined as the general problem of compensating the photometrical disparities between two coarsely geometrically registered images. In other words, color correction consists of transferring the color palette of a reference image, usually called source image (**S**), to a target image (**T**) [19].

There are several color correction approaches in the literature, however, the majority of them focus on correcting a pair of images, whereas our objective is to increase the level of similarity of multiple images. Color correction approaches, in a general sense, can be divided into two main classes: model-based parametric and model-less non-parametric.

Model-based parametric approaches assume that the color distribution of the images follows some statistical distribution. In 2001, Reinhard et al. [20] used simple Gaussians to model the color distributions of images **S** and **T**, proposing the following linear transformation:

$$\hat{f}(x) = \bar{y} + \frac{V_y}{V_x} \cdot (x - \bar{x}), \quad (1)$$

where  $(\bar{y}, V_y)$  represent the mean and standard variation of the global color distribution of **S** in the  $l\alpha\beta$  color space,  $(\bar{x}, V_x)$  represent the mean and standard variation of the global color distribution of **T** in the  $l\alpha\beta$  color space, and  $x$  corresponds to the color of a pixel in **T**. This work was one of the first model-based parametric methods for color correction, and is commonly used as baseline method for comparing other color correction approaches [21–23]. Additional research focused on improving the modeling of the color distributions of the images, due to the simplicity of Gaussian distributions. For instance, in [24], the authors divided the images into regions, then matched the regions of both images, and for each match of regions applied a similar method to the one proposed in [20]. In addition, the authors also proposed the usage of Gaussian Mixture Models (GMMs) and the Expectation Maximization (EM) algorithm to model the color distribution more accurately. In [25], the authors first divided the images into spatially connected regions using the mean shift algorithm, then each region was modeled as a collection of truncated Gaussians using a maximum likelihood estimation procedure, in order to model more accurately the local color distributions. In [22], the authors proposed an extension of [20] to the RGB color space, by the development of a simple statistic-based method that uses both the mean and covariance matrix. Another problem identified by [20] was the appearance of cross channel artifacts caused by the modelling and correcting of each channel of the images independently. In [23], the authors tried to solve this problem by using 3D GMMs to model the color distribution across the 3 channels of the images. Other model-based parametric approaches use gain compensation techniques [26–28], only operating in the intensity channel rather than in the full color space.

Model-less non-parametric approaches make no assumptions about the nature of the color distributions of the images. Usually, in these approaches, a 2D Joint Image Histogram (JIH) is computed from the overlapped areas of two images [19]. The JIH is then used to estimate a Color Mapping Function (CMF). In this process, there are two points that deserve to be highlighted: a) robust methods are usually required in order to deal with the outliers caused by different illumination, camera exposure variation [29], vignetting effect [30], reflection properties of certain objects, capturing angles, among others [19]; b) the monotonicity property of the CMF has to be maintained [31]. There are several examples of

model-less non-parametric approaches, and what differentiates most of the approaches is how they tackle those two. For example, in [31], the authors propose a robust estimation of the Brightness Transfer Function (BTF) from the JIH of two overlapped images. The authors also used dynamic programming to enforce the monotonicity constraint in the estimation of the BTF. In [32,33], the authors solved these two issues in a separated way: first they used a 2D tensor voting to remove the noise and produce an initial estimation of the CMF; then they used a heuristic local adjustment method to regulate the CMF estimation and enforce the monotonicity increasing constraint. The same authors, in [34], experimented the usage of a Bayesian framework to estimate the CMF. In [35], the authors proposed an energy minimization method in the histogram to estimate a robust CMF, while in [36], high-dimensional Bezier patches were used to estimate the CMF. In [37], the authors employed the moving least squares framework with spatial constraints to correct images in the RGB color space. In [29], the authors proposed the usage of a root-polynomial regression that is invariant to the camera exposure variation and scene irradiance. Another option, as seen in [38], is to use the Vandermonde matrix to find an interpolating polynomial.

Color correction approaches can also be divided based on whether they are global or local. Global color correction approaches [20,22,39,40] assume that estimating a single color mapping function for the entire image should be sufficient to perform a proper color correction. In complex scenes, global color correction approaches are often unable to accurately model the color distribution in the images due to factors such as color clusters, differing optics, sensor characteristics, hardware processing employed by video cameras, among others [41]. Local color correction approaches [24,25,31,42,43] go further by segmenting the image into spatially connected regions or color clusters, and then fitting a color mapping function to each of the obtained segments. These approaches are able to model the color distribution more precisely, and, in most cases, this translates to a more accurate color correction process. However, local approaches usually do not allow a direct mapping of color between pixels for a pair of images because they are limited by the accuracy of the geometric registration. As a result, there is either a maximum number or a minimum area of segmented regions for the image [25]. In other words, for very small regions or, in the degenerate case, single pixel regions, the color correction will fail because of misalignment in the region mappings, caused by error in the geometric registration. Also, statistical inference over small sample sizes is noisy and there are local color correction problems such as image clarity reduction and structural inconsistency. To tackle these issues, [44] proposed the optimization of a residual image between the target image and the corrected one. There are also authors, such as [33,45], who have tried to combine benefits of both approaches by taking into consideration global and local color information.

The two most complete performance evaluations of color correction approaches were carried out in [19] and in [25]. In [19], the authors tested nine color correction algorithms applied to 70 pairs of images. The conclusions of that evaluation state that, in the model-based parametric approaches, gain compensation [28] and local color transfer [24] were the approaches that yielded the best results, when considering the color transfer effectiveness metric. In the model-less non-parametric approaches, the best approaches, considering that same evaluation metric, were the tensor voting in joint image space [33] and brightness transfer function [31]. Furthermore, the authors claim that, using more evaluation metrics, such as extendability, stability, and speed, both the model-based parametric approaches cited above [24,28] achieved the best results. In [25], the authors carried out a performance evaluation of 11 different color correction algorithms, applied to 63 pairs of images and used three evaluation metrics. They demonstrated that their approach, modeling of regions using truncated Gaussians, outperforms all other 10 approaches considering the PSNR and S-CIELAB metrics. Furthermore, the authors show that, with respect to color correction performances, the RGB and  $l\alpha\beta$  color spaces achieve similar results.

The previous lines have detailed several approaches to cope with the problem of color correcting a pair of images. Texture mapping applications often have several images of the scene, and this is one of the shortcomings of the state of the art: most approaches

are designed to operate using only two images, and it is not straightforward how those methods could be expanded to accommodate multiple images. 155

Another relevant issue is the usage of spatial information to enhance the color correction. There are several global approaches [20,22,24], which do not use spatial information as an input to the color correction procedure. However, since they do not require spatial registration between the images, their effectiveness is limited. Some other proposed approaches make use of spatial information, but only at the image level, meaning that they require the images to be geometrically registered with respect to each other. As seen in [19,31], and more recently [25], the usage of this additional information enhances the effectiveness of the color correction with respect to that of global approaches. However, there is additional information that could further enhance the color correction procedure, which has not been used thus far. We refer to the spatial registration, not only between the images in the scene, but also with respect to the range measurements of that scene (the point clouds) and the mesh that represents the surfaces of the scene. Using this data, it is possible to detect occlusions and other situations that may lead to poor performance of the color correction algorithm. 160 165

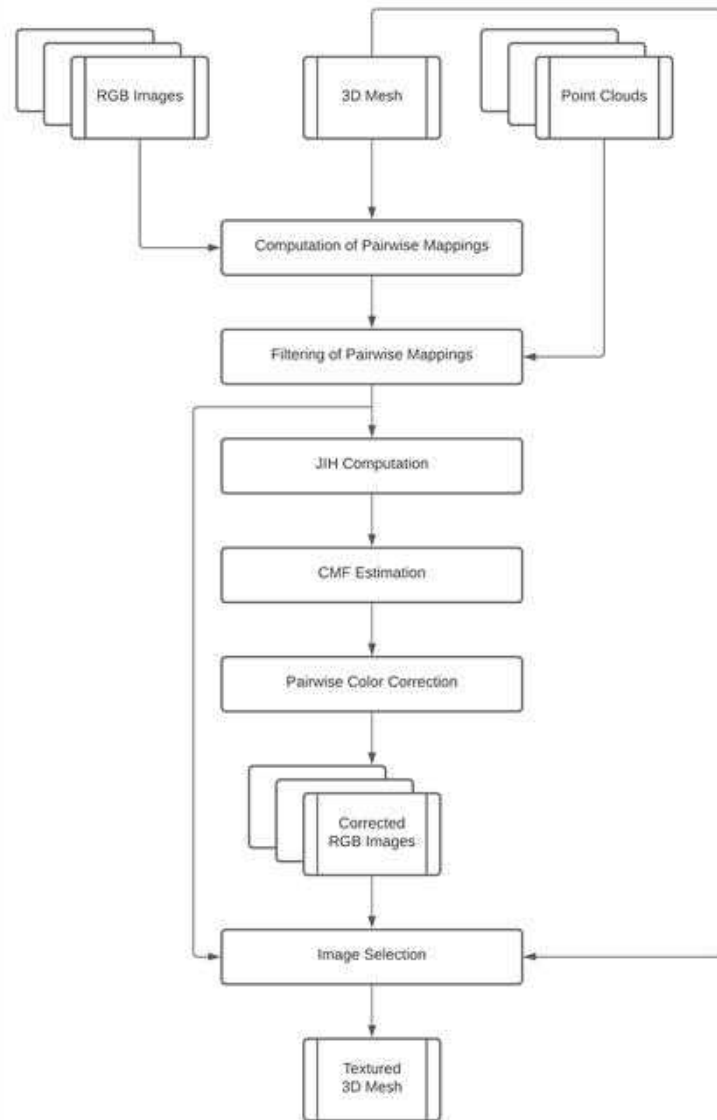
This paper proposes a novel method that attempts to make use of that additional information in order to improve the effectiveness of the color correction procedure. Furthermore, we proposed a pairwise-based methodology that is able to tackle the problem of color correcting multiple images. The procedure works by color correcting all images in the dataset against a common source image, referred to as reference image. We have also implemented two distinct image selection algorithms to assess the performance of the proposed color correction algorithms under the application of different camera selection criteria. 170 175

The remainder of the paper is structured as follows: in Section 2, the proposed approach is presented, Section 3 discusses the results, and Section 4 presents the conclusions.

## 2. Proposed Approach 180

The architecture of the proposed approach is depicted in Figure 1. As input, our system takes RGB images, registered point clouds from different scans, and a 3D mesh obtained from those point clouds. All RGB images must be geometrically registered with respect to the 3D model, given that our main objective is to use 3D information to enhance the similarity between the images. Firstly, the faces present in the 3D mesh are projected onto the images, to create what we call pairwise mappings, i.e., for each of the mesh triangles, we compute the pixel coordinates of the projection corresponding to the three vertices in each pair of images. Then, we apply several techniques to filter the noisy pairwise mappings that would undermine both the color correction process and the texture mapping. After a successful removal of the noisy pairwise mappings, we compute a Joint Image Histogram (JIH) for every pair of images and estimate a Color Mapping Function (CMF) that best fits the JIH data. To finish the color correction procedure, we perform pairwise color correction using the CMFs created in the previous step, effectively correcting all images with respect to a chosen image that must be carefully selected (reference image). At the end of this step, we produce the corrected RGB images. To analyze the influence of this 3D-based color correction in 3D meshes, we use the corrected images to colorize each face of the 3D mesh using 2 different techniques that will be explained later on. Besides using color corrected images to colorize the 3D mesh, we also make use of the information from the pairwise mappings filtering component to increase the robustness of the image selection technique. At the end, we produce the textured mesh. 185 190 195 200

In the remainder of this section, we describe every component of the architecture and present the intermediate results for each of the components. The dataset (Figure 2) used to produce the intermediate results was taken from a laboratory, containing 7 images, 9 point clouds from different scans of the entire room, and a 3D mesh with 46,627 faces and 27,454 vertices from a corner of the room. 205



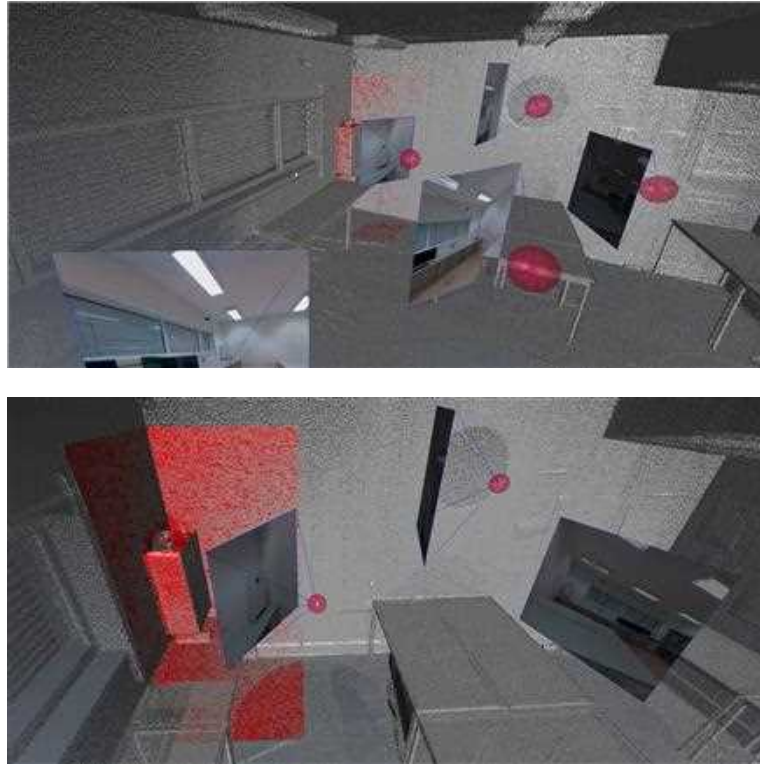
**Figure 1.** Architecture of the proposed approach.

### 2.1. Computation of Pairwise Mappings

The first step of the 3D-based color correction is the projection of the mesh faces onto the images. For this purpose, we use the pinhole camera model [46] to compute the projections of all vertices of the faces. The projection of the 3D coordinates ( $\mathbf{x}_{f,v} \in \mathbb{R}^3$ ) of the  $v$ -th vertex of the  $f$ -th face, of the set of faces  $\mathcal{F}$ , onto the  $i$ -th image ( $\mathbf{x}_{f,v}^{(i)} \in \mathbb{Z}^2$ ), of the set of images  $\mathcal{I}$ , can be described as:

$$\begin{aligned} \mathbf{x}_{f,v}^{(i)} &= \mathbf{K} \cdot \mathbf{T}_i \cdot \mathbf{x}_{f,v}, \\ \forall f \in \mathcal{F} \wedge \forall v \in \{1, 2, 3\} \wedge \forall i \in \mathcal{I}, \end{aligned} \quad (2)$$

where  $\mathbf{T}_i$  represents the extrinsic parameters associated with the camera that took the  $i$ -th image ( $3 \times 4$  matrix) and  $\mathbf{K}$  represents the camera's intrinsic parameters ( $3 \times 3$  matrix). Since all images were taken using the same camera,  $\mathbf{K}$  will be considered the same for all images. Throughout the document, we use the  $[\cdot]^{(i)}$  notation to denote a projection onto the  $i$ -th image. It is important to note that, although we are projecting the vertices of the mesh onto the images, our atomic units are faces and not vertices. Thus, to assess the validity



**Figure 2.** Two viewpoints from the dataset. The gray points represent the point cloud, the red surface represents the 3D mesh, and the red spheres represent the scan locations. The images are also shown, with frustum in blue.

of the projection of the  $f$ -th face onto the  $i$ -th image, two conditions must be fulfilled: all vertices of the  $f$ -th face must be projected inside the  $i$ -th image and the  $z$  component of the 3D coordinates of the  $f$ -th face vertices, transformed to the camera's coordinate reference system, must be greater than 0. This assessment can be formulated as: 220

$$\mathcal{F}^{(i)} = \left\{ f \in \mathcal{F} : 0 \leq [\mathbf{x}_{f,v}^{(i)}]_x < W \wedge 0 \leq [\mathbf{x}_{f,v}^{(i)}]_y < H \wedge [\mathbf{T}_i \cdot \mathbf{x}_{f,v}]_z > 0 \right\}, \quad (3)$$

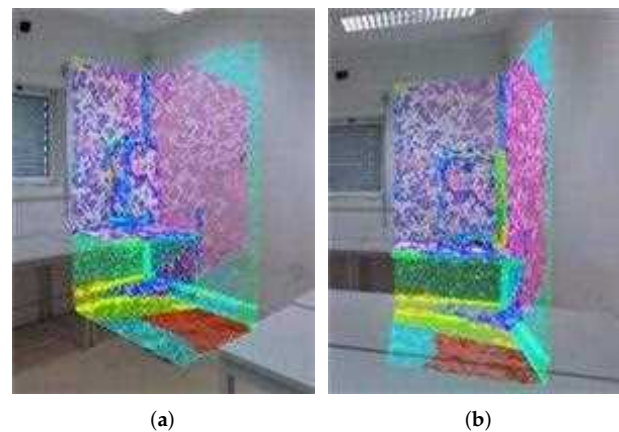
$$\forall v \in \{1, 2, 3\},$$

where  $\mathcal{F}^{(i)}$  is a set that contains the faces with valid projection to the  $i$ -th image,  $W, H$  represents the width and height of the image,  $[\cdot]_x$ ,  $[\cdot]_y$ , and  $[\cdot]_z$  are operators that extract the  $x$ ,  $y$ , and  $z$  coordinates, respectively.

After the calculation of the valid projections, we compute what we refer to as pairwise mappings. Pairwise mappings are pixel coordinates from a pair of images that correspond to the same projected vertex. Firstly, we loop through each face for every pair of images, and evaluate if the projection of that face is valid, considering both images. Then, we create a pairwise mapping for each vertex that consists of a tuple with two elements  $\left( \left\langle \mathbf{x}_{f,v}^{(i)}, \mathbf{x}_{f,v}^{(k)} \right\rangle \right)$ : the pixel coordinates of the vertex projection onto the first image and onto the second image. This procedure can be formulated as: 225

$$\mathcal{M}_{(i,k)} = \left\{ \left\langle \mathbf{x}_{f,v}^{(i)}, \mathbf{x}_{f,v}^{(k)} \right\rangle : f \in \mathcal{F}^{(i)} \wedge f \in \mathcal{F}^{(k)} \right\}, \forall v \in \{1, 2, 3\}, \quad (4)$$

where  $\mathcal{M}_{(i,k)}$  is the set that contains the pairwise mappings associated with the  $i$ -th and  $k$ -th images. In [Figure 3](#), the pairwise mappings are illustrated by using the same color to showcase correspondence mappings between images. 230



**Figure 3.** Pairwise mappings. Points are colored to identify the mappings between images. For instance, in both images, the points in pink belong to the wall.

We believe that the usage of pairwise mappings is more accurate and reliable than the information computed based on the overlapping areas of two images, since it contains photometrical information between pixels that correspond to the same projected vertex, instead of containing photometrical information between regions of pixels that may or may not correspond to the same 3D object. However, the state of the pairwise mappings at this stage of the pipeline is far from optimal, because they contain noisy data caused by the problems mentioned in Section 1. In the next section, such problems and the proposed solutions for them will be discussed.

## 2.2. Filtering of Pairwise Mappings

The pairwise mappings computed as described in Section 2.1 contain a significant amount of noise, as can be seen in Figure 3. For example, the red colored pixels should correspond to the same projected 3D object, which is untrue because in Figure 3(a) those pixels are the projection of the ground, while in Figure 3(b) those pixels are the projection of the table surface. Inaccurate pairwise mappings are caused by two reasons: occluded faces and registration errors. The most common source of noise in our system are faces occluded in at least one of the images. This happens when two conditions are met: a) the occluding face intersects the line of sight between the camera and the occluded face; b) the occluding face is closer to the camera. The usage of pairwise mappings with occluded faces introduces incorrect photometrical data into the system, which undermines the color correction procedure. To tackle this problem, we use two filtering methods, as in [14]: z-buffering and depth consistency. Although these filtering methods aim to tackle the same problem, their scope of work is different, as will be explained below. Regarding the registration errors, we are using professional equipment that guarantees a low average registration error, and therefore, we do not propose solutions to reduce it. Still, even though the average registration error may be low, in situations where the camera viewpoint with respect to the face is excessively oblique, i.e, the angle between the focal axis of the camera and the normal vector of the face is near  $90^\circ$ , the error impact on the accuracy of the pairwise mappings is amplified, potentially leading to inaccurate pairwise mappings. For this reason, we propose an additional filtering method, which we call camera viewpoint filtering, that focuses on discarding pairwise mappings of faces that have an excessive oblique angle with respect to the camera focal axis.

In order to eliminate the noisy pairwise mappings, we propose a filtering procedure composed of a combination of the three filtering methods described above: z-buffering filtering, depth consistency filtering, and camera viewpoint filtering. Each pairwise mapping computed using Equation 4 is submitted to this procedure, and is only considered valid if it passes the 3 filtering methods. In the next paragraphs, each of the individual

falterings will be analyzed in detail, introducing the equations that allow their computation. The evolution of the pairwise mappings throughout the filtering procedure is depicted in [Figure 5](#), where it is possible to visualize which pairwise mappings are affected by each filtering method.

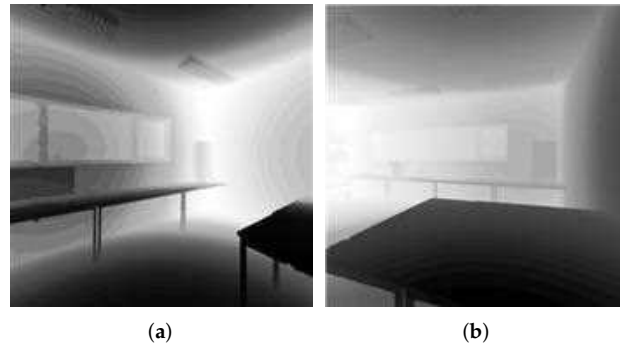
The z-buffering filtering method can be formulated as:

$$\mathcal{F}^{(i)} = \left\{ f \in \mathcal{F} : \neg f^{(i)} \cap g^{(i)} \vee \left( f^{(i)} \cap g^{(i)} \wedge \max_v \left( \left\| \mathbf{T}_i \cdot \mathbf{x}_{f,v} \right\| \right) < \min_v \left( \left\| \mathbf{T}_i \cdot \mathbf{x}_{g,v} \right\| \right) \right) \right\}, \quad (5)$$

$$\forall g \in \mathcal{F} \setminus f, \forall v \in \{1, 2, 3\},$$

where  $\mathcal{F}^{(i)}$  is the set that contains the faces that are not occluded when considering the  $i$ -th image,  $f^{(i)}$  and  $g^{(i)}$  represent the projections of the  $f$ -th and  $g$ -th faces onto the  $i$ -th image, respectively, and  $\|\cdot\|$  denotes the  $L_2$  norm. In summary, to assert whether the  $f$ -th face is occluded using the z-buffering filter, a comparison with respect to all other faces must be carried out. To conclude that the  $f$ -th face is not occluded by the  $g$ -th face, when considering the  $i$ -th image, one of two conditions must be met: the intersection between the projection of the vertices of the  $f$ -th face ( $f^{(i)}$ ) and the projection of the vertices of the  $g$ -th face ( $g^{(i)}$ ) must be the empty set; or, in case the intersection is different than the empty set, the maximum euclidean distance of the three vertices of the  $f$ -th, relative to the camera, must be less than the minimum euclidean distance of the three vertices of the  $g$ -th, face relative to the camera. [Figure 5\(c\)](#) and [Figure 5\(d\)](#) contain the pairwise mappings between the same images shown in [Figure 3](#), this time with the z-buffering filter applied. It is possible to observe that in [Figure 5\(d\)](#) the mappings on the right side of the boiler (zone limited by the black rectangle) were discarded because in [Figure 5\(c\)](#) they are occluded by the boiler itself. Note that the red colored mappings were not discarded by z-buffering, because this method can only discard pairwise mappings due to occlusions when both the occluded object and occluding object are represented in the 3D mesh. As detailed previously, we only have the 3D mesh of the corner of the room, meaning that the table does not exist in the 3D mesh. To tackle this variant of the occlusion, we propose the second filtering method - depth consistency. This method aims to discard occlusions due to discrepancies between the mesh and the point cloud. These discrepancies can have multiple sources, such as mesh irregularities, non-defined mesh, registration issues, among others. The main idea of depth consistency is to use information both from the 3D mesh and from the point clouds to estimate the depth of the faces via 2 different paths, and then conclude, based on the difference of those values, whether depth inconsistency is present. Note that, by depth, we mean distance from the camera to the center of the face. When the depth values are inconsistent, we can assume that the face considered does not belong to the object that is being viewed by the camera. Therefore, using this mapping would introduce incorrect photometric mappings into the system. Our approach can be divided into two parts: a) depth computation of the  $f$ -th face based on the 3D mesh; b) depth computation of the  $f$ -th face based on the point cloud. The depth computation based on the 3D mesh is carried out by computing the  $L_2$  norm of the vertices coordinates with respect to the camera, and then selecting the minimum value. To perform the depth computation based on the point cloud, we first use the partial point clouds from the setups where the images were taken and create depth images. An example of these depth images can be seen in [Figure 4](#). Once the depth images are computed, the depth can be directly extracted using the coordinates obtained from vertex projection. The depth consistency filtering can be represented as:





**Figure 4.** Depth images. The whiter the pixel, the more distant the projected point is relative to the camera. For visualization purposes, these images are the result of an equalization in order to enhance the contrast.

$$\mathcal{F}^{(i)} = \left\{ f \in \mathcal{F} : \left| \overbrace{\min_v \left( \left\| \mathbf{T}_i \cdot \mathbf{x}_{f,v} \right\| \right)}^{\text{mesh}} - \overbrace{\min_v \left( \mathbf{D}^{(i)} \left( \begin{bmatrix} \mathbf{x}_{f,v}^{(i)} \end{bmatrix}_y, \begin{bmatrix} \mathbf{x}_{f,v}^{(i)} \end{bmatrix}_x \right) \right)}^{\text{point cloud}} \right| < t_{dc} \right\}, \quad (6)$$

$$\forall v \in \{1, 2, 3\},$$

where  $\mathcal{F}^{(i)}$  is the set that contains the selected faces using the depth consistency filter when considering the  $i$ -th image,  $\mathbf{D}^{(i)}$  corresponds to the depth image associated with the  $i$ -th image, and  $t_{dc}$  is the depth consistency threshold, establishing the maximum admissible discrepancy between the distances computed using these two different methods. 315

Figure 5(e) and Figure 5(f) contain the pairwise mappings with both the z-buffering filter and the depth consistency filter applied. By comparing them with Figure 5(c) and Figure 5(d), two differences should be highlighted: a) the mappings inside the blue rectangle in Figure 5(f) were discarded because they are occluded by the left leg of the table in Figure 5(e). Although one could expect those mappings to be removed with the z-buffering, that is not the case, since the leg of the table was not defined on the mesh, due to its reduced thickness, making z-buffering ineffective in this case. However, since the depth consistency filter compares the depth between the mesh and the point cloud, it was able to detect the discrepancy; b) the mappings inside the brown rectangle in Figure 5(f) were discarded because the 3D mesh was not defined in that region, and as we can see, that is not a problem, since the depth consistency filter can detect and eliminate those false pairwise mappings. 320

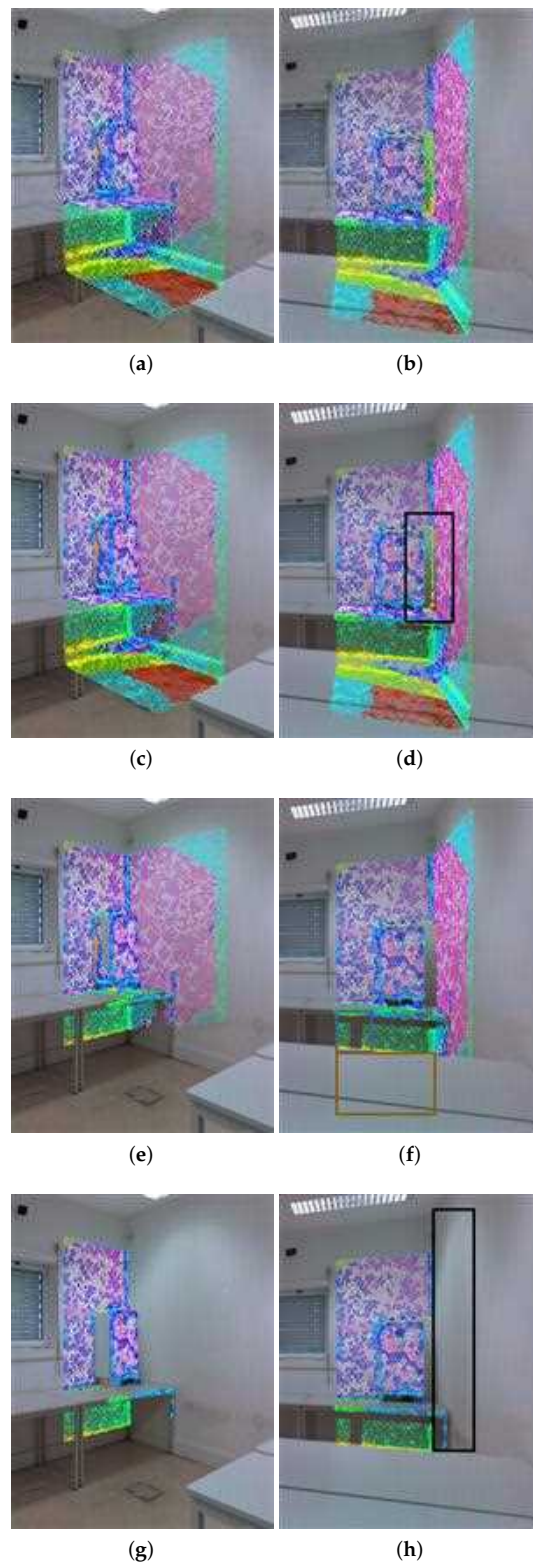
Lastly, we propose a method to remove pairwise mappings in which the face has an excessively oblique viewpoint relative to the camera. We call this method camera viewpoint filtering. The assumption behind the removal of these pairwise mappings is that when the camera viewpoint is excessively oblique with respect to the face, the impact the registration error has on the accuracy of the pairwise mappings is amplified, potentially leading to the use of incorrect photometric information. Furthermore, this method allows us to discard mappings between the front and rear surfaces of a face, similarly to what is done in the back face culling method [47]. Let  $\vec{f}_i$  be the unit vector with the direction from the  $i$ -th camera to the center of the  $f$ -th face, and  $\vec{N}_f$  be the unit vector, normal to that face. The camera viewpoint filtering can be described as: 325

$$\mathcal{F}^{(i)} = \{ f \in \mathcal{F} : \vec{f}_i \cdot \vec{N}_f < t_{cv} \}, \quad (7)$$

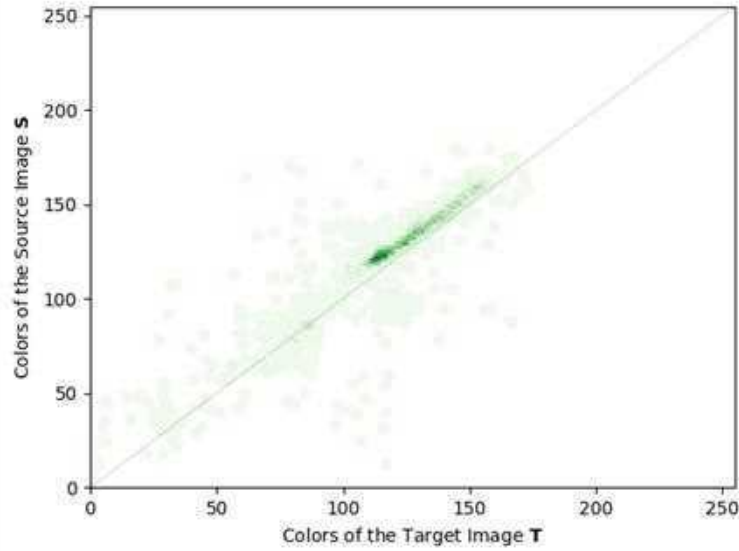
where  $\mathcal{F}^{(i)}$  is the set that contains the faces that passed the camera viewpoint filter when considering the  $i$ -th image and  $t_{cv}$  is a threshold that defines the admissible angle between  $\vec{f}_i$  and  $\vec{N}_f$ . 340

Figure 5(g) and Figure 5(h) contain the pairwise mappings with all filtering methods applied (z-buffering filtering, depth consistency filtering, and camera viewpoint filtering). Regarding the camera viewpoint filtering, the most significant difference is inside the zone marked with a black rectangle in Figure 5(h), where the pairwise mappings that belong to faces that were being captured in an excessively oblique viewpoint were discarded. 345

In the present case study, this filtering procedure led to the removal of 74.23% of the pairwise mappings, which clearly demonstrates the very large amount of noise in this dataset. This reinforces the importance of using 3D information to discard incorrect pairwise mappings. Furthermore, this filtering procedure can have a huge impact in the image selection, because it allows us to avoid coloring a face with an image in which that face is occluded or with an excessively oblique viewpoint. 350



**Figure 5.** Pairwise mappings filtering procedure: (a) and (b) pairwise mappings with no filter applied; (c) and (d) pairwise mappings with z-buffering applied; (e) and (f) pairwise mappings with z-buffering filter and depth consistency applied; and (g) and (h) pairwise mappings with the entire filtering procedure applied.



**Figure 6.** Example of the green channel of a JIH using the filtered pairwise mappings.

### 2.3. Joint Image Histogram Computation

In this step we compute a Joint Image Histogram (JIH) for each pair of images, using the filtered pairwise mappings. Let  $\mathbf{x}^{(i)}$ ,  $\mathbf{y}^{(k)}$  be the pixel coordinates of the mappings in the target image ( $\mathbf{T}$ ), and source image ( $\mathbf{S}$ ), respectively, and  $\mathbf{I}(\mathbf{x}^{(i)})$ ,  $\mathbf{I}(\mathbf{y}^{(k)})$  denote the color intensity of the  $\mathbf{x}^{(i)}$  and  $\mathbf{y}^{(k)}$  pixel coordinates in the  $i$ -th image and  $k$ -th image, respectively. The JIH is built using the following equation:

$$\text{JIH}(\mathbf{x}, \mathbf{y}) = \sum_{\langle \mathbf{x}^{(i)}, \mathbf{y}^{(k)} \rangle \in \mathcal{M}} \delta(\mathbf{x}, \mathbf{I}(\mathbf{x}^{(i)})) \cdot \delta(\mathbf{y}, \mathbf{I}(\mathbf{y}^{(k)})), \quad (8)$$

where  $\delta(\cdot)$  is the Kronecker delta function, and  $\mathbf{x}$  and  $\mathbf{y}$  represent all possible values of colors in  $\mathbf{T}$  and  $\mathbf{S}$ , being defined in the discrete interval  $[0, 2^n[$ , where  $n$  is the bit depth of the images. Figure 6 shows an example of a JIH of the green channel. The green dots represent all the observations according to the mappings for that pair of images. A histogram is an observation count, which is represented by the color intensity of each point. The JIHs are used to estimate the CMFs,  $\hat{f}_m$ , which will then be used for producing the color corrected images.

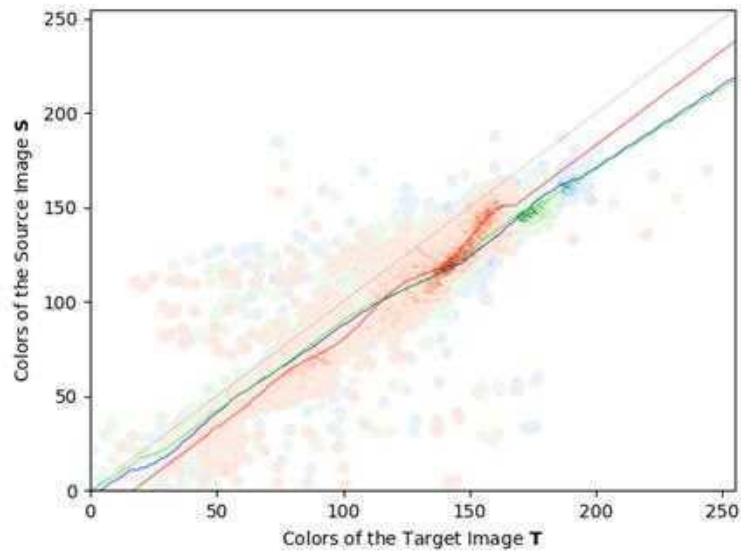
### 2.4. Estimation of Color Mapping Functions

In this step, the current paper proposes to use the JIH as a base to estimate a Color Mapping Function (CMF), which are functions that map the colors of a source image  $\mathbf{S}$  to a target image  $\mathbf{T}$  resulting in a color corrected image  $\hat{\mathbf{T}}$ . In this context, the CMF can be expressed as:

$$\hat{\mathbf{x}} = \hat{f}_m(\mathbf{x}), \quad \forall \mathbf{x} \in [0, 2^n[, \quad (9)$$

where  $\hat{f}_m$  is the estimated CMF for each of the pairwise images, according to the mappings created, and  $\hat{\mathbf{x}}$  is the resultant color of the color corrected image  $\hat{\mathbf{T}}$  for a given color  $\mathbf{x}$  of the target image  $\mathbf{T}$ .

Since the CMF is a function, it cannot map a single element of its domain to multiple elements of its codomain. For the entire range of values (according to the bit depth of the image) for each color of the target image ( $\mathbf{x}$ ) there is one and only one resulting corrected value ( $\hat{\mathbf{x}}$ ). However, a typical JIH has several observations of  $\mathbf{y}$  for each of the values in  $\mathbf{x}$ ,



**Figure 7.** Example of the estimation of three Color Mapping Functions, one for each channel.

therefore it must be assumed that the JIH contains a set of considerably noisy observations. Also, it is important to note that, as there are three JIH for every pair of images, one for each channel, there are also three CMFs.

The current paper proposes to estimate the CMF using a regression analysis to fit the observations from the JIH. We utilize a regression analysis called Support Vector Regression (SVR) [48], making use of the Linear Kernel and the Radius Basis Function Kernel to estimate the single CMF approach used in the pipeline being presented. [Figure 7](#) shows an example of the CMF estimation for a given JIH of a pair of images. The red, green and blue curves represent the CMFs estimation. For each channel, the function appears to fit the data of the JIH quite well, following the greater peaks of observations.

### 2.5. Pairwise Color Correction

In this step, we propose to perform what we call pairwise color correction by choosing a reference camera and using the CMF estimated previously for that given pair of images ( $T, S$ ). The image from the reference camera will be the source image  $S$  and will form pairs with all the other images as target images  $T$ . Since the selection of the reference camera is arbitrary, a high quality image should be selected.

Even though the CMF was estimated from a JIH created using only the filtered pairwise mappings, the current paper proposes the generalization of the CMF to color correct all pixels of the target image  $T$ . This generalization increases the similarity between all the images, not only within the mappings, but in the entire image, showing that our color correction approach can handle multiple images.

For each pair of images there are three color mapping functions that were estimated previously, one for each channel. That means that each of the three channels will be color corrected independently. To perform the color correction, the color value  $x$  of each pixel of the target image  $T$  will be given as input to the CMF and then replaced by the color corrected value  $\hat{x}$ , according to [Equation 9](#). After performing the color correction for each channel independently, the channels are merged to generate the corrected image  $\hat{T}$ .

[Figure 8](#) shows an example for a pair of images. As we can observe, the target image  $T$  became more similar to the source image  $S$ , indicating that the color correction was successful. [Figure 9](#) shows the effectiveness of the proposed approach to handle the color correction of multiple images and increase their similarity. The objective of the current paper is to evaluate the quality of the color correction by examining, not only the similarity



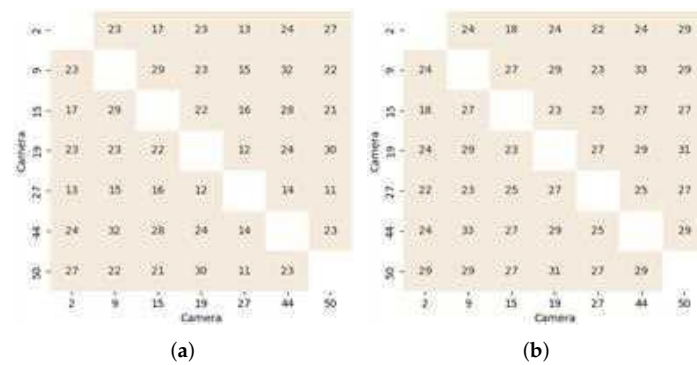
**Figure 8.** Example of the color correction for a pair of images: (a) Chosen reference camera (source image  $S$ ); (b) Original target image  $T$ ; (c) Color corrected target image  $\hat{T}$ .

of the images with respect to each other, but by analyzing the success of their use in the improvement of the textured mesh quality. 410

### 2.6. Image Selection

The last step to produce the textured mesh is what we refer to as image selection. The goal is to select, for each face, the image used to colorize that face. This process of colorization is based on cropping, from the selected image, the projection of the face. We propose to use, not only the corrected images to create a higher-quality mesh, but also the information associated with the filtering of pairwise mappings (Section 2.2). To select the image to colorize the  $f$ -th face, in addition to the information provided by the filtering pairwise mappings, we need a criteria to select one image from the available images. We implemented two different criteria: random selection and selection based on the largest area of the projection. The random approach selects a random image from the available ones. To analyze the impact of the color correction in the textured mesh, this is the best approach because it amplifies the perception of the dissimilarities between all images. The selection based on the largest area of the projection consists of selecting the image where the area of the face projection is larger. This method produces higher-quality textured meshes, but it is less suitable to evaluate the impact of the color correction because it produces meshes with less image transitions between adjacent faces. 415  
420  
425

[Figure 10\(a\)](#) and [Figure 10\(d\)](#) showcase the result of image selection using the random selection and largest projection area selection, respectively. In [Figure 10\(a\)](#) it is possible to observe that the random selection appears to be a more suitable image selection technique to assess the effectiveness of the color correction procedure, since it uses several images to colorize the faces within the same surfaces, leading to the creation of multiple texture seams artefacts. While the largest projection area selection tries to use the same image to colorize all faces on a certain surface, the number of zones where a transition of the selected image exists is reduced, decreasing the likelihood of texture seams artefacts, and with it, 430



**Figure 9.** Confusion Matrices using PSNR metric which the higher the score, the more similar the pair of images are: (a) Original images (no color correction); (b) Corrected images with proposed approach.

the impact of color correction. [Figure 10\(b\)](#) and [Figure 10\(c\)](#) represent the textured mesh produced with the random selection technique, using the original and corrected images, respectively. The positive impact of the color corrected images on the textured mesh is considerable, eliminating most of the visual unappealing artefacts, and thus creating a more visually appealing textured mesh. [Figure 10\(e\)](#) and [Figure 10\(f\)](#) contain the same structure as [Figure 10\(b\)](#) and [Figure 10\(c\)](#), but showcase the image selection based on the largest area of the projection. We can observe that the quality of the textured mesh using the color corrected images has improved, but since the number of transitions between images is smaller, the improvement is less noticeable. Nonetheless, the overall appearance of the textured mesh is highly affected by the smoothness of transition between images, and as we can see in [Figure 10](#), our proposed color correction approach produces smoother transitions, leading to higher-quality textured meshes.



**Figure 10.** Textured meshes produced. (a) and (d) illustrate the result of the random selection and the selection based on the largest area of the projection, respectively. The color of each face is associated with the selected image, meaning that faces with the same color have the same selected image. The resultant textured mesh with random selection as the image selection technique is depicted in (b) using the original images, and in (c) using the color corrected images. The resultant textured mesh with image selection based on the largest area of the projection is depicted in (e) using the original images, and in (f) using the color corrected images

### 3. Results

Section 2 presented some preliminary results to illustrate each of the stages of the pipeline; this section contains the complete results, showcasing the color corrected images and their impact on the visual quality of the meshes from different viewpoints. Firstly, a comparison between the proposed approach and other state-of-the-art color correction algorithms is presented. This was divided into three sections, in order to analyze images and textured meshes. Section 3.1 and Section 3.2 present image-based quantitative and qualitative evaluations, respectively. Section 3.3 presents mesh-based qualitative evalua-



**Table 1.** Algorithms compared with the proposed approach.

| Algorithm | Reference                    | Description                                 |
|-----------|------------------------------|---|
| #1        | -                            | Baseline (target image)                     |
| #2        | Reinhard et al. [20]         | Global Color Transfer                       |
| #3        | Reinhard et al. adapted [20] | Global Color Transfer – Lightness Channel   |
| #4        | Finlayson et al. [29]        | Root-Polynomial Regression Color Correction |
| #5        | De Marchi et al. [38]        | Vandermonde Matrices for Color Correction   |
| #6        | This paper                   | 3D-based Color Correction                   |

tions. In Section 3.4 and Section 3.5 we discuss the impact of the proposed filtering methods, and of the selection of the reference image, on the quality of the color correction process. 455

Table 1 lists all the evaluated algorithms. Algorithm #1 is the baseline approach, which means that no color correction is performed. Algorithm #2 is a global approach proposed in [20], which uses a linear transformation (Equation 1) to impose the global color distribution of a source image onto a target image, in the  $l\alpha\beta$  color space. Algorithm #3 is an adaptation of Algorithm #2, in which only the lightness channel of the  $l\alpha\beta$  color space is corrected (luminance correction). Algorithm #4 is the approach proposed in [29], which uses a root-polynomial regression to estimate a CMF. Algorithm #5 uses the Vandermonde matrix to compute the coefficients of a polynomial [38] that is applied as a CMF. In both #4 and #5, the unfiltered pairwise mappings are used to produce the color correspondence vectors required by these approaches. 460 465

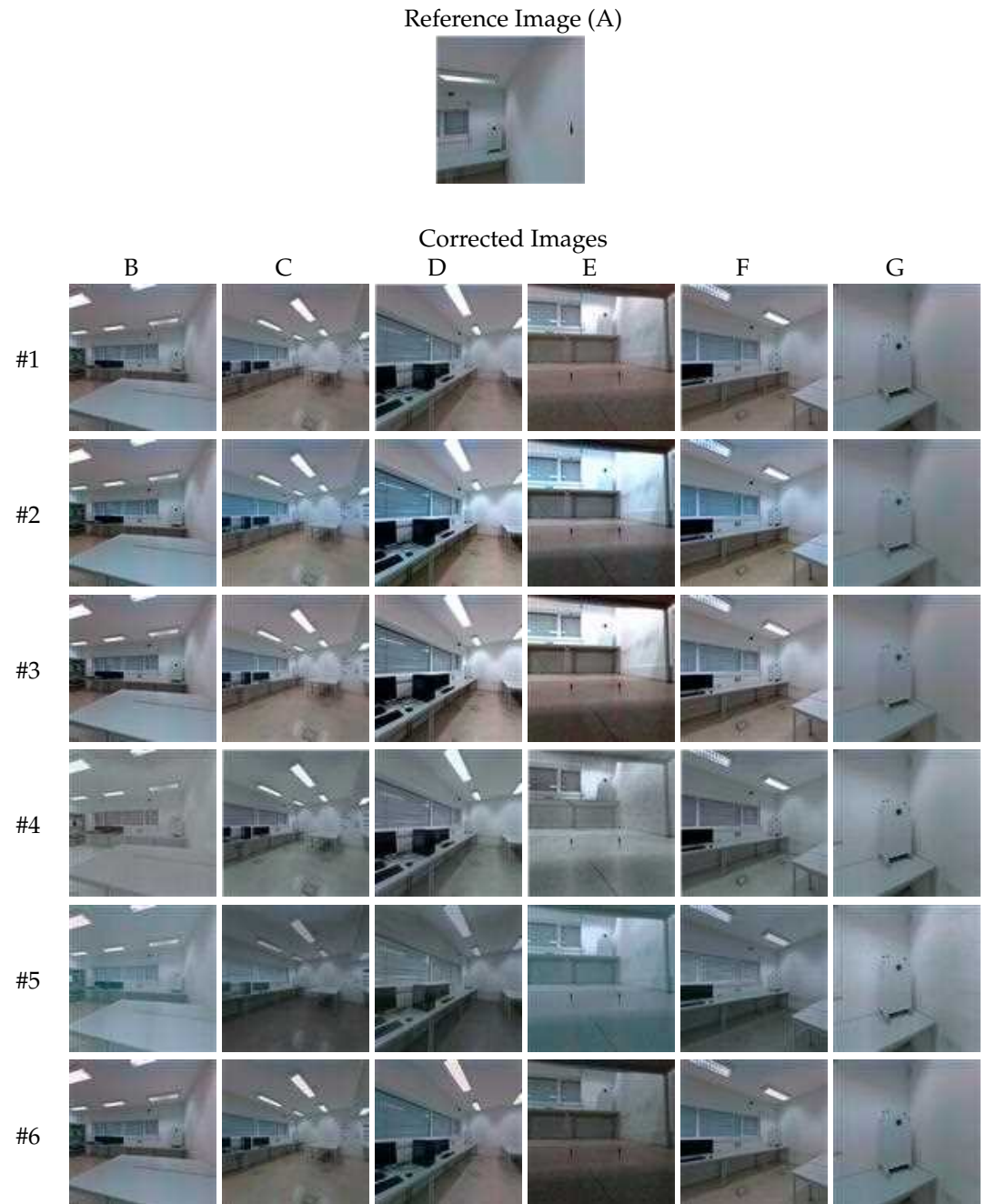
Section 2.5 presented the pairwise color correction, where a reference image is selected as source to form pairs with the remaining images as target images. Since the dataset contains 7 images, 6 pairs of images are formed to be directly color corrected, and we refer to those pairs of images as Subset 1. Additionally, in order to analyze the capacity of the approaches to increase the color similarity between all the images, a Subset 2 is formed, composed by image pairs that were not directly color corrected, meaning neither of the images was the source in the color correction procedure. Subset 2 is therefore composed of 15 pairs of images, obtained from the combination of the available images, excluding the reference image. These two subsets combined form the complete dataset, which contains 21 pairs of images. These groups of images will be used for the quantitative evaluations. 470 475

The quantitative performance evaluation of each color correction algorithm is determined employing two different image similarity metrics over the filtered pairwise mappings. The usage of the metrics solely with the filtered pairwise mappings is the best way to fairly compare the robustness of each algorithm because it allows us to analyze the overlapping portions of each pair of images without being influenced by the abundance of noise present before filtering. The first metric, proposed in [19], is known as peak signal-to-noise ratio (PSNR). It is a similarity measure, so the higher the score values, the more similar the pair of images are. The second metric is named CIEDE2000. It is a dissimilarity metric, meaning that more similar images result in lower score values. This metric is adopted as the most recent color difference metric from the International Commission on Illumination (CIE), and was improved in [49]. 480 485

### 3.1. Image-Based Qualitative Evaluation

A comparison between the approaches (Table 1) is presented by qualitative evaluation of the color similarity between images. As described in Section 2.5, the pairwise color correction is performed in 6 pairs of images (Subset 1) that were created with the same reference image. After the procedure, the target images should have become more similar in color to the reference image (image A in this case) and, as a consequence, more similar with respect to each other as well. Table 2 shows the color corrected images produced by each approach. In the first row (algorithm #1), the target images B through G are shown in their initial form (uncorrected). Afterwards, global approaches #2 and #3 seem to have excessively increased the brightness of the images. Between the two, #3 appears to have performed a marginally better color correction. For algorithm #4, the corrected images have 490 495

**Table 2.** Corrected images B, C, D, E, F, G (columns) using image A as reference image (source image). Each row corresponds to a different color correction algorithm.



become more similar to each other. However, the overall color palette obtained appears quite different from the reference image. The images in #5 are quite degenerated, more noticeably images C, D and F. From the analysis of the obtained results, the most similar images in color to the reference image were produced by the proposed approach (#6). In image E (algorithm #6), the reduction of brightness is evident on the left side of the boiler, which improved the color similarity with both the reference image and the other images. It is noteworthy that, even though the pairwise mappings were created only for the boiler zone, the generalization of the CMFs in the other parts of the image achieved very good color correction results as well.

500

505

**Table 3.** Mean and Standard Deviations of the PSNR and CIEDE2000 (CIED) scores for each algorithm in Subset 1, Subset 2, and Dataset. Best results highlighted in bold.

| Alg. | Subset 1     |          |             |          | Subset 2     |          |             |          | Dataset      |          |             |          |
|------|--------------|----------|-------------|----------|--------------|----------|-------------|----------|--------------|----------|-------------|----------|
|      | PSNR         |          | CIED        |          | PSNR         |          | CIED        |          | PSNR         |          | CIED        |          |
|      | $\mu$        | $\sigma$ | $\mu$       | $\sigma$ | $\mu$        | $\sigma$ | $\mu$       | $\sigma$ | $\mu$        | $\sigma$ | $\mu$       | $\sigma$ |
| #1   | 22.33        | 5.31     | 8.42        | 4.90     | 21.00        | 6.20     | 9.11        | 5.30     | 21.38        | 6.00     | 8.91        | 5.19     |
| #2   | 19.05        | 5.60     | 11.10       | 6.63     | 16.39        | 5.09     | 14.17       | 6.68     | 17.15        | 5.38     | 13.29       | 6.81     |
| #3   | 19.00        | 5.69     | 10.83       | 6.26     | 16.49        | 5.31     | 13.24       | 6.72     | 17.21        | 5.54     | 12.55       | 6.68     |
| #4   | 23.00        | 4.83     | 6.97        | 3.85     | 21.47        | 5.31     | 7.50        | 3.83     | 21.90        | 5.23     | 7.35        | 3.84     |
| #5   | 20.83        | 3.72     | 8.83        | 4.37     | 19.07        | 5.54     | 11.07       | 6.12     | 19.57        | 5.15     | 10.43       | 5.77     |
| #6   | <b>27.50</b> | 5.92     | <b>3.48</b> | 0.39     | <b>26.60</b> | 4.08     | <b>4.09</b> | 1.21     | <b>26.86</b> | 3.71     | <b>3.91</b> | 1.08     |

### 3.2. Image-Based Quantitative Evaluation

In this section, an evaluation of the similarity between images using the **PSNR** and **CIEDE2000** quantitative metrics is presented. **Table 3** shows the results separately for the following sets: Subset 1, Subset 2, and the complete dataset. Global approaches #2 and #3 could not deal with the distinct color clusters in order to compute a good global color distribution, even with the slight improvement obtained through the adaptation made in algorithm #3. Algorithms #4 and #5 were the second and third best approaches, respectively, considering both image similarity metrics (**PSNR** and **CIEDE2000**) in all sets. The proposed approach outperformed all the other algorithms, in all sets, and in both metrics. Concerning the **PSNR** similarity metric, the proposed approach achieved averages scores of 27.50, 26.60, and 26.86 in Subset 1, Subset 2, and Dataset, respectively. For the **CIEDE2000** dissimilarity metric, the proposed approach achieved average scores of 3.48, 4.09, and 3.91 in Subset 1, Subset 2, and Dataset, respectively.

As expected, Subset 1 achieved the best overall scores because it contains only pairs of images that were directly color corrected. It is worth noting that Subset 2 also achieved very good scores, confirming the capacity of the proposed approach to increase the color similarity between multiple images. The only algorithm besides the proposed approach that achieved better overall results than the baseline approach was algorithm #4. These results reinforce the notion that, to accomplish a successful color correction, it is not only important to estimate better CMFs, with high-level techniques, but also to effectively reduce the amount of noise in the input data, caused by registration errors, occluded surfaces, incorrect photometric correspondences, among others.

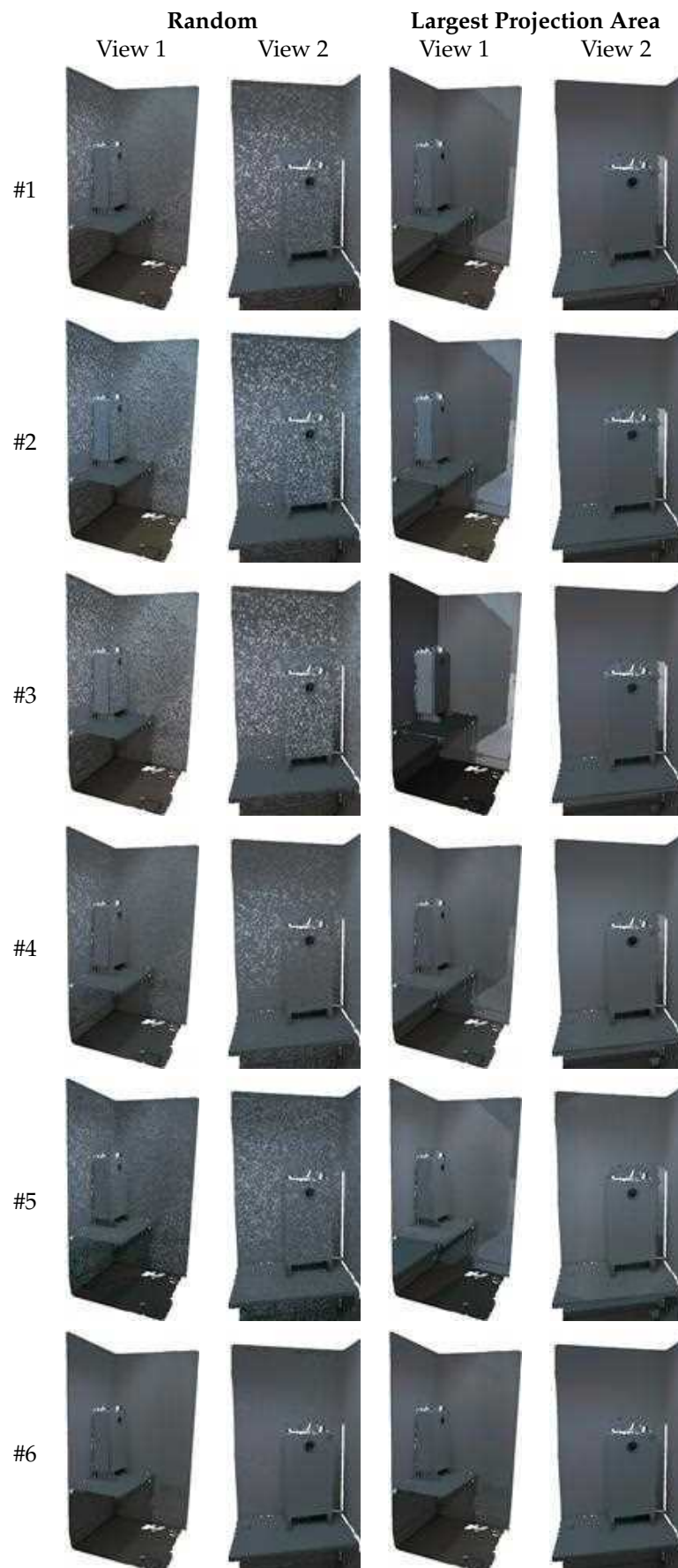
### 3.3. Mesh-Based Qualitative Evaluation

In this section, a comparison between the approaches (**Table 1**) is presented by evaluating the visual quality of textured meshes created using two different image selection criteria: random selection and largest projection area selection. In order to achieve a fair evaluation, all the meshes produced by each approach have exactly the same texture mapping for both images selection techniques. As discussed in Section 2.6, the usage of multiple images to texture a mesh will result in the transition between images to map different areas of the mesh. The apparent color difference at the limits between neighboring triangles creates visual artifacts, known as texture seams. **Figure 10(a)** and **Figure 10(d)** illustrate where these transitions are present in the mesh, using representative colors. **Table 4** presents the textured meshes for each approach, using the two image selection techniques, in two different viewpoints. The first viewpoint shows an overview of the entire mesh. The second viewpoint shows a zoomed-in region of the boiler and the wall.

Since algorithm #1 uses the original images, the visual artifacts created by the changes in the selected image are noticeable. The random image selection criterion aggravates the effect of this problem, but it can be mitigated by color correcting the images. The meshes produced by the approaches #2 through #5 appear unable to improve the visual quality

of the textured mesh, presenting a large amount of very noticeable texture seams. The proposed approach (algorithm #6) appears to produce the highest quality textured mesh.

Even when using the random image selection criterion, the texture seams are less noticeable, verifying that indeed the images become more similar in color with respect to each other. Moreover, with the largest projection area selection criterion, the visual quality of the mesh raises even more, displaying almost no texture seams. These results demonstrate the robustness of the proposed approach by showing a significant improvement in the visual quality of the textured mesh.

**Table 4.** Textured meshes produced by each approach from two different viewpoints and using two different image selection techniques: random selection and largest projection area selection.

**Table 5.** Variants of the proposed approach used to evaluate the impact of the filtering methods.

| Algorithm       | Description                                    |
|-----------------|--|
| #6 <sup>a</sup> | Proposed approach without any filtering method |
| #6 <sup>b</sup> | Proposed approach using only z-buffering       |
| #6 <sup>c</sup> | Proposed approach using only depth consistency |
| #6 <sup>d</sup> | Proposed approach using only camera viewpoint  |
| #6              | Proposed approach                              |

### 3.4. Impact of the Proposed 3D-Based Filtering Methods

In this section, the objective is to evaluate the impact each filtering method has on the color correction outcome. For that, we evaluate four variants of the proposed approach, listed in [Table 5](#).

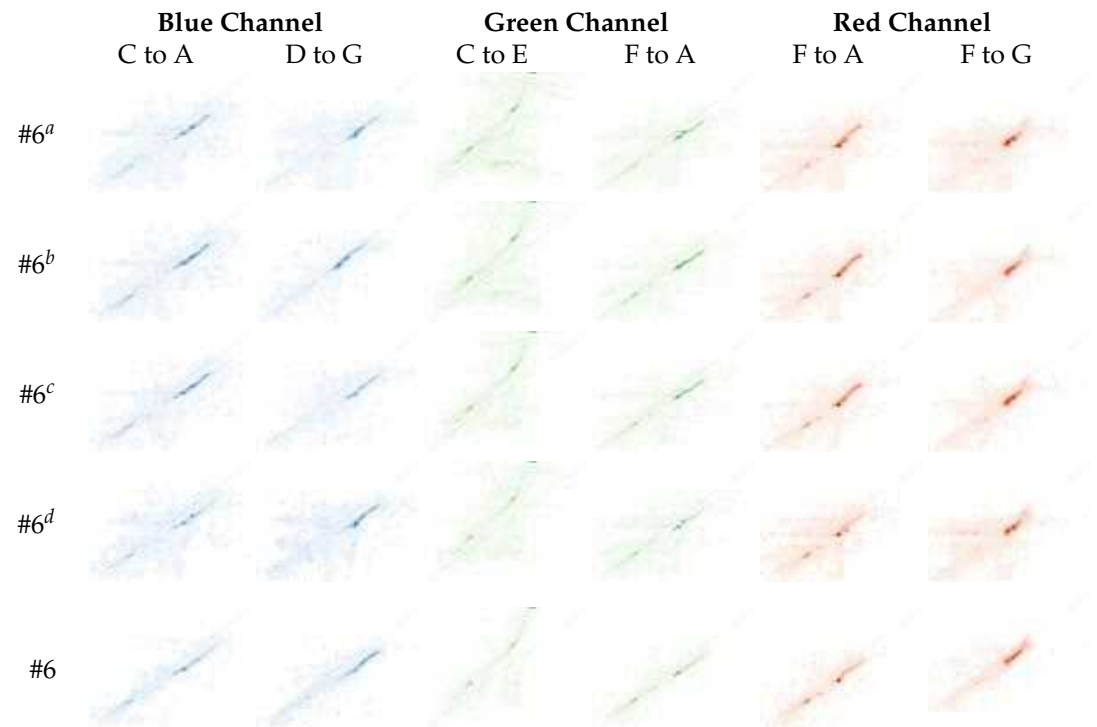
We propose to evaluate the impact of each filtering method from two perspectives: assessing the percentage of pairwise mappings removed by analyzing the JIH ([Table 6](#) and [Table 7](#)); using image similarity metrics ([Table 8](#)).

[Table 6](#) depicts the average percentage of pairwise mappings removed when considering all available image pairs (Dataset). Algorithm #6 removed 74.23% of the pairwise mappings, demonstrating the importance of using robust techniques to handle noise. Through an individual analysis of each filtering method, we can note that algorithm #6<sup>d</sup> was the component that resulted in the largest reduction of the pairwise mappings. On the other hand, algorithm #6<sup>b</sup>, was the one with the least impact on the pairwise mapping removal. [Table 7](#) illustrates the impact that each approach has on the JIH. As was expected, due to the 74.23% reduction of the pairwise mappings, algorithm #6 completely transforms the JIH when compared to the JIH produced using the unfiltered pairwise mappings (algorithm #6<sup>a</sup>). The JIHs produced by algorithm #6 contain less correspondences of colors in the source image for each color in the target image, which is exactly what we expect from a noise free JIH. In other words, algorithm #6 is able to produce thinner JIHs, which enable the estimation of more accurate CMFs, leading to higher levels of color similarity between images. The JIH depicted in cell (#6, FA), in the green channel, is a very good approximation of a noiseless JIH. Based on the conclusions extracted from [Table 6](#), it was expected that the individual filtering approach that should have the most impact on the transformation of the JIH was algorithm #6<sup>d</sup>, which is not the case. The reason for this is that the majority of the pairwise mappings removed by algorithm #6<sup>d</sup> belong to zones where the JIH is denser, and therefore the difference is not highly noticeable. This is an interesting conclusion because it shows that to evaluate the effectiveness of a filtering method, in addition to assessing the number of mappings removed, an analysis of the zones of the JIH where those mappings are must be carried out. The individual filtering approach that produced thinner JIHs was algorithm #6<sup>c</sup>, and the cells where that impact is the most significant are (#6<sup>c</sup>, FA), in the green channel, and (#6<sup>c</sup>, FG), in the red channel. Although in this dataset, the most significant filter was algorithm #6<sup>c</sup>, it is important to note that the effectiveness of each filter depends on the characteristics of the dataset used.

**Table 6.** Average percentage of pairwise mappings removed by each algorithm.

| Alg.            | $\mu$ | $\sigma$ |
|-----------------|-------|----------|
| #6 <sup>a</sup> | 0.00  | 0.00     |
| #6 <sup>b</sup> | 26.27 | 13.06    |
| #6 <sup>c</sup> | 39.71 | 7.45     |
| #6 <sup>d</sup> | 56.41 | 15.17    |
| #6              | 74.23 | 10.81    |

**Table 7.** JIHs produced using the different variants of the proposed approach (rows). Each column represents one color channel of the JIH between a randomly chosen pair of images. For example, the first column, C to A, represents the blue channel of the JIH between image C and image A.



**Table 8** shows the quantitative results achieved by all algorithms in the same sets as in **Table 3**: Subset 1, Subset 2, and Dataset. As expected, the pairs of images in Subset 1 present a higher level of color similarity than the pairs of images in Subset 2. For the Subset 1, algorithm #6 achieved the best results both in the **PSNR** metric and **CIEDE2000** metric, with average scores of 27.50 and 3.48, respectively. For the Subset 2, when considering the **PSNR** metric, algorithm #6<sup>b</sup> achieved an average score of 26.86, followed by algorithm #6 - 26.60. Regarding the **CIEDE2000** metric, algorithm #6 was, once again, the approach that achieved the best results, with 4.09. Finally, when considering the entire dataset, algorithm #6<sup>b</sup> achieved an average score of 27.00 in the **PSNR** metric, and in the **CIEDE2000**, the best results belong to algorithm #6, with an average score of 3.91. **CIEDE2000** takes into account the sensibility of human-eye to certain colors [49], making this image similarity metric more suitable for image-based comparisons, and therefore, we consider that algorithm #6 was the approach that achieved the best results. It is noteworthy that both algorithm #6<sup>b</sup> and algorithm #6<sup>c</sup> in all sets achieved remarkable results, being quite close to the score achieved by algorithm #6.

From these results, we can conclude that the usage of the 3D-based filtering methods is a significant contribution to the quality of the color correction procedure.

**Table 8.** Mean and standard deviation of the **PSNR** and **CIEDE2000 (CIED)** scores for each algorithm

| Alg.            | Subset 1     |          |             |          | Subset 2     |          |             |          | Dataset      |          |             |          |
|-----------------|--------------|----------|-------------|----------|--------------|----------|-------------|----------|--------------|----------|-------------|----------|
|                 | PSNR         |          | CIED        |          | PSNR         |          | CIED        |          | PSNR         |          | CIED        |          |
|                 | $\mu$        | $\sigma$ | $\mu$       | $\sigma$ | $\mu$        | $\sigma$ | $\mu$       | $\sigma$ | $\mu$        | $\sigma$ | $\mu$       | $\sigma$ |
| #6 <sup>a</sup> | 26.50        | 2.99     | 4.27        | 0.77     | 26.26        | 3.97     | 4.80        | 1.47     | 26.33        | 3.72     | 4.65        | 1.33     |
| #6 <sup>b</sup> | 27.33        | 2.49     | 3.83        | 0.46     | <b>26.86</b> | 3.70     | 4.28        | 1.28     | <b>27.00</b> | 3.41     | 4.16        | 1.13     |
| #6 <sup>c</sup> | 27.00        | 2.77     | 3.95        | 0.53     | 26.53        | 4.06     | 4.35        | 1.41     | 26.67        | 3.75     | 4.24        | 1.24     |
| #6 <sup>d</sup> | 26.33        | 3.14     | 4.22        | 1.03     | 26.20        | 4.02     | 4.93        | 1.54     | 26.24        | 3.79     | 4.72        | 1.45     |
| #6              | <b>27.50</b> | 5.92     | <b>3.48</b> | 0.39     | 26.60        | 4.08     | <b>4.09</b> | 1.21     | 26.86        | 3.71     | <b>3.91</b> | 1.08     |

**Table 9.** Mean and standard deviation of the PSNR, CIEDE2000 (CIED) scores for each reference image selection, using algorithm #6.

| Ref. | Subset 1     |          |             |          | Subset 2     |          |             |          | Dataset      |          |             |          |
|------|--------------|----------|-------------|----------|--------------|----------|-------------|----------|--------------|----------|-------------|----------|
|      | PSNR         |          | CIED        |          | PSNR         |          | CIED        |          | PSNR         |          | CIED        |          |
|      | $\mu$        | $\sigma$ | $\mu$       | $\sigma$ | $\mu$        | $\sigma$ | $\mu$       | $\sigma$ | $\mu$        | $\sigma$ | $\mu$       | $\sigma$ |
| A    | 27.50        | 5.92     | 3.48        | 0.39     | 26.60        | 4.08     | <b>4.09</b> | 1.21     | <b>26.86</b> | 3.71     | <b>3.91</b> | 1.08     |
| B    | 23.50        | 3.25     | 4.55        | 1.73     | <b>27.40</b> | 2.65     | 4.19        | 0.81     | 26.29        | 3.34     | 4.29        | 1.16     |
| C    | 27.67        | 3.68     | 3.48        | 0.78     | 25.20        | 3.58     | 4.60        | 1.54     | 25.90        | 3.78     | 4.28        | 1.46     |
| D    | 25.33        | 4.50     | 4.55        | 2.06     | 25.33        | 3.81     | 4.91        | 1.29     | 25.33        | 4.02     | 4.81        | 1.55     |
| E    | 21.33        | 1.97     | 6.28        | 0.86     | 22.27        | 3.53     | 7.96        | 2.76     | 22.00        | 3.19     | 7.48        | 2.49     |
| F    | 28.00        | 3.21     | <b>3.38</b> | 0.66     | 25.07        | 3.66     | 4.47        | 1.40     | 25.90        | 3.78     | 4.19        | 1.36     |
| G    | <b>29.00</b> | 1.15     | 4.00        | 0.55     | 25.67        | 3.94     | 5.20        | 1.98     | 26.62        | 3.71     | 4.86        | 1.78     |

### 3.5. Impact of the Selection of Reference Image on Image-Based Quantitative Evaluation

One of the hyperparameters that must be defined in the proposed approach is the selection of the reference image from all available images. The goal of this section is to evaluate the impact that this selection has on the color correction outcome. Table 9 contains the quantitative results achieved when selecting each image as the reference. Note that, although the reference image is different, all other configurations are the same. For Subset 1, image G achieved the best average score concerning the PSNR metric (29.00), while in the CIEDE2000 metric, the reference image that attained the best average score was image F, with a value of 3.38. Regarding Subset 2, the reference image that achieved the best average score when considering the PSNR metric was image B (27.40), and concerning the CIEDE2000 metric, image A achieved an average score of 4.09. Finally, regarding the entire Dataset, image A achieved the best results in both the PSNR metric and CIEDE2000 metric, with average scores of 26.86 and 3.91, respectively. These results demonstrate that the effectiveness of the color correction is not strongly dependent on the selection of the reference image. The results using image E as the reference image achieved the worst performance in both evaluation metrics when considering all sets. It is noteworthy that, using this image, the results for the PSNR metric in Subset 1 were subpar to those obtained in Subset 2. This can be explained by the fact that image E was taken under a table, see Table 2, which caused a different illumination from the rest of the images, hampering the color correction process. Still, algorithm #6 using image E as the reference still outperforms algorithms #2, #3 and #5, achieving equivalent results to algorithm #4, see Table 3. Any other image used as reference for the proposed approach produces results that outperform the other evaluated algorithms.

## 4. Conclusions

This paper proposed two main contributions to the state of the art of color correction: a pairwise-based methodology capable of color correcting multiple images from the same scene; the usage of 3D information to filter out noisy mappings between pairs of images. The pairwise-based methodology consists of selecting one image as the reference image and then color correcting all other images with respect to the reference image. This procedure increases, not only the color similarities of all images with respect to the reference image, but also the color similarities between all image in the dataset. The usage of 3D information consists of applying 3D information from the real world, namely point clouds and meshes, to build a filtering procedure capable of producing more reliable spatial registration between images, which results in more accurate color correction.

Results demonstrate that the proposed approach outperforms all other approaches, both using qualitative and quantitative metrics. Furthermore, results show that our color correction procedure enhances the visual quality of the produced textured meshes. We tested two different image selection criteria, random selection and largest area projection selection, and demonstrated that even using the most simple one - random selection, the



proposed approach produces textured meshes virtually free of texture seams. In addition, we explored how each filtering method affects both the JIH and the color correction outcome, where we demonstrated that all filtering methods play an important role in the color correction procedure. Using the filtering tools that we propose, it is also possible to quantify the amount of noise that exists in datasets used for color correction, which in our case was around 75%. This value is quite large, reinforcing the thesis of our work: robust filtering techniques are crucial for successful color correction algorithms. In the end, we performed a comparative analysis, where we show that the effectiveness of our proposed approach is not strongly dependent on the selected reference image.

For future work, we plan to explore several techniques to estimate color mapping functions in order to take full advantage of the filtered pairwise mappings that we obtain using the filtering procedure proposed in this paper. In addition, we intend to integrate other color correction algorithms with our filtering procedure and evaluate the impact that our filtering procedure has on the color correction outcome. We also plan to test our color correction procedure in scenarios with a wider color range, in order to analyze its robustness concerning different scenarios.

**Author Contributions:** Conceptualization, P.D. and M.O.; Funding acquisition, P.D. and M.O.; Methodology, D.C., L.D., P.D. and M.O.; Software, D.C., L.D., T.M. and M.O.; Writing – original draft, D.C. and L.D.; Writing – review editing, T.M., P.D. and M.O. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work has been partially supported by Portugal 2020, project IDT n° 45382 in collaboration with VR360, and by FCT - Portuguese Foundation for Science and Technology, in the context of Ph.D. scholarship 2020.07345.BD.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Ziegler, R.; Matusik, W.; Pfister, H.; McMillan, L. 3D reconstruction using labeled image regions. *Proceedings of the Eurographics Symposium on Geometry Processing* **2003**, pp. 248–260.
2. Ham, H.; Wesley, J.; Hendra, H. Computer Vision Based 3D Reconstruction : A Review. *International Journal of Electrical and Computer Engineering (IJECE)* **2019**, *9*, 2394. doi:10.11591/ijece.v9i4.pp2394-2402.
3. Pérez, L.; Rodríguez, Í.; Rodríguez, N.; Usamentiaga, R.; García, D.F. Robot Guidance Using Machine Vision Techniques in Industrial Environments: A Comparative Review. *Sensors* **2016**, *16*. doi:10.3390/s16030335.
4. Zhang, Y.; Chen, H.; Waslander, S.L.; Yang, T.; Zhang, S.; Xiong, G.; Liu, K. Toward a More Complete, Flexible, and Safer Speed Planning for Autonomous Driving via Convex Optimization. *Sensors* **2018**, *18*. doi:10.3390/s18072185.
5. Trinidad-Fernández, M.; Beckwée, D.; Cuesta-Vargas, A.; González-Sánchez, M.; Moreno, F.A.; González-Jiménez, J.; Joos, E.; Vaes, P. Validation, Reliability, and Responsiveness Outcomes of Kinematic Assessment with an RGB-D Camera to Analyze Movement in Subacute and Chronic Low Back Pain. *Sensors* **2020**, *20*. doi:10.3390/s20030689.
6. Di Angelo, L.; Di Stefano, P.; Guardiani, E.; Morabito, A.E.; Pane, C. 3D Virtual Reconstruction of the Ancient Roman Incile of the Fucino Lake. *Sensors* **2019**, *19*. doi:10.3390/s19163505.
7. Vázquez-Arellano, M.; Griepentrog, H.W.; Reiser, D.; Paraforos, D.S. 3-D Imaging Systems for Agricultural Applications—A Review. *Sensors* **2016**, *16*. doi:10.3390/s16050618.
8. Marroquim, R.; Pfeiffer, G.; Carvalho, F.; Oliveira, A.A.F. Texturing 3D models from sequential photos. *The Visual Computer* **2012**, *28*, 983–993.
9. Liu, L.; Nguyen, T.Q. A Framework for Depth Video Reconstruction From a Subset of Samples and Its Applications. *IEEE Transactions on Image Processing* **2016**, *25*, 4873–4887.

10. Neugebauer, P.J.; Klein, K. Texturing 3D Models of Real World Objects from Multiple Unregistered Photographic Views. *Computer Graphics Forum* **2001**, *18*, 245–256. 700
11. Guo, Y.; Wan, J.; Zhang, J.; Xu, K.; Lu, M. Efficient registration of multiple range images for fully automatic 3D modeling. 2014 International Conference on Computer Graphics Theory and Applications (GRAPP), 2014, pp. 1–8.
12. Callieri, M.; Cignoni, P.; Corsini, M.; Scopigno, R. Masked Photo Blending: mapping dense photographic dataset on high-resolution 3D models. *Computer & Graphics* **2008**, *32*, 464–473.
13. Kehl, W.; Navab, N.; Ilic, S. Coloured signed distance fields for full 3D object reconstruction. Proceedings of the British Machine Vision Conference. BMVA Press, 2014. 705
14. Oliveira, M.; Lim, G.H.; Madeira, T.; Dias, P.; Santos, V. Robust texture mapping using rgb-d cameras. *Sensors* **2021**, *21*, 1–22. doi:10.3390/s21093248.
15. Lempitsky, V.S.; Ivanov, D.V. Seamless Mosaicing of Image-Based Texture Maps. *2007 IEEE Conference on Computer Vision and Pattern Recognition* **2007**, pp. 1–6. 710
16. Allène, C.; Pons, J.P.; Keriven, R. Seamless image-based texture atlases using multi-band blending. *2008 19th International Conference on Pattern Recognition* **2008**, pp. 1–4.
17. Gal, R.; Wexler, Y.; Ofek, E.; Hoppe, H.; Cohen-Or, D. Seamless Montage for Texturing Models. *Eurographics 2010* **2010**, 29/2.
18. Pérez, P.; Gangnet, M.; Blake, A. Poisson Image Editing. *ACM Trans. Graph.* **2003**, *22*, 313–318.
19. Xu, W.; Mulligan, J. Performance evaluation of color correction approaches for automatic multi-view image and video stitching. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* **2010**, pp. 263–270. doi:10.1109/CVPR.2010.5540202. 715
20. Reinhard, E.; Ashikhmin, M.; Gooch, B.; Shirley, P. Color transfer between images. *IEEE Computer Graphics and Applications* **2001**, *21*, 34–41. doi:10.1109/38.946629.
21. Zhang, M.; Georganas, N.D. Fast color correction using principal regions mapping in different color spaces. *Real-Time Imaging* **2004**, *10*, 23–30. doi:10.1016/j.rti.2003.11.001. 720
22. Xiao, X.; Ma, L. Color transfer in correlated color space. *Proceedings - VRCA 2006: ACM International Conference on Virtual Reality Continuum and its Applications* **2006**, pp. 305–309. doi:10.1145/1128923.1128974.
23. Oliveira, M.; Sappa, A.D.; Santos, V. Color correction for onboard multi-camera systems using 3D Gaussian mixture models. *IEEE Intelligent Vehicles Symposium, Proceedings* **2012**, pp. 299–303. doi:10.1109/IVS.2012.6232141. 725
24. Tai, Y.W.; Jia, J.; Tang, C.K. Local color transfer via probabilistic segmentation by expectation-maximization. *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005* **2005**, *1*, 747–754. doi:10.1109/CVPR.2005.215.
25. Oliveira, M.; Sappa, A.D.; Santos, V. A probabilistic approach for color correction in image mosaicking applications. *IEEE Transactions on Image Processing* **2015**, *24*, 508–523. doi:10.1109/TIP.2014.2375642.
26. Nanda, H.; Cutler, R. Practical calibrations for a real-time digital omnidirectional camera. *CVPR Technical Sketch* **2001**. 730
27. Brown, M.; Lowe, D.G. Recognising panoramas. *Proceedings of the IEEE International Conference on Computer Vision* **2003**, *2*, 1218–1225. doi:10.1109/iccv.2003.1238630.
28. Brown, M.; Lowe, D.G. Automatic Panoramic Image Stitching Automatic 2D Stitching. *International Journal of Computer Vision* **2007**, *74*, 59–73.
29. Finlayson, G.D.; Mackiewicz, M.; Hurlbert, A. Color Correction Using Root-Polynomial Regression. *IEEE Transactions on Image Processing* **2015**, *24*, 1460–1470. doi:10.1109/TIP.2015.2405336. 735
30. Zheng, Y.; Lin, S.; Kambhamettu, C.; Yu, J.; Kang, S.B. Single-image vignetting correction. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2009**, *31*, 2243–2256. doi:10.1109/TPAMI.2008.263.
31. Kim, S.J.; Pollefeys, M. Robust radiometric calibration and vignetting correction. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2008**, *30*, 562–576. doi:10.1109/TPAMI.2007.70732. 740
32. Jia, J.; Tang, C.K. Image registration with global and local luminance alignment. *Proceedings of the IEEE International Conference on Computer Vision* **2003**, *1*, 156–163. doi:10.1109/iccv.2003.1238331.
33. Jia, J.; Tang, C.K. Tensor voting for image correction by global and local intensity alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2005**, *27*, 36–50. doi:10.1109/TPAMI.2005.20.
34. Jia, Y.; Sun, J.; Tang, C.K.; Shum, H.Y. Bayesian correction of image intensity with spatial consideration. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **2004**, *3023*, 342–354. doi:10.1007/978-3-540-24672-5\_27. 745
35. Yamamoto, K.; Oi, R. Color correction for multi-view video using energy minimization of view networks. *International Journal of Automation and Computing* **2008**, *5*, 234–245. Cited By :41.
36. Sajadi, B.; Lazarov, M.; Majumder, A. ADICT: Accurate direct and inverse color transformation. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **2010**, *6314 LNCS*, 72–86. doi:10.1007/978-3-642-15561-1\_6. 750
37. Hwang, Y.; Lee, J.Y.; Kweon, I.S.; Kim, S.J. Probabilistic moving least squares with spatial constraints for nonlinear color transfer between images. *Computer Vision and Image Understanding* **2019**, *180*, 1–12. doi:10.1016/j.cviu.2018.11.001.
38. De Marchi, S. Polynomials arising in factoring generalized Vandermonde determinants: an algorithm for computing their coefficients. *Mathematical and Computer Modelling* **2001**, *34*, 271–281. doi:https://doi.org/10.1016/S0895-7177(01)00060-7. 755

39. Pitie, F.; Kokaram, A.; Dahyot, R. N-dimensional probability density function transfer and its application to color transfer. Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1. IEEE, 2005, Vol. II, pp. 1434–1439 Vol. 2. doi:10.1109/ICCV.2005.166.
40. Pitié, F.; Kokaram, A.C.; Dahyot, R. Automated colour grading using colour distribution transfer. *Computer Vision and Image Understanding* **2007**, *107*, 123–137. doi:10.1016/j.cviu.2006.11.011. 760
41. Yin, J.; Cooperstock, J. Color correction methods with applications to digital projection environments. *Wscg* **2004**, pp. 499–506.
42. Xiang, Y.; Zou, B.; Li, H. Selective color transfer with multi-source images. *Pattern Recognition Letters* **2009**, *30*, 682–689. doi:10.1016/j.patrec.2009.01.004.
43. Oliveira, M.; Sappa, A.D.; Santos, V. Unsupervised local color correction for coarsely registered images. CVPR 2011. IEEE, 2011, pp. 201–208. doi:10.1109/CVPR.2011.5995658. 765
44. Niu, Y.; Liu, P.; Zhao, T.; Fan, Y. Matting-based residual optimization for structurally consistent image color correction. *IEEE Transactions on Circuits and Systems for Video Technology* **2020**, *30*, 3624–3636. doi:10.1109/TCSVT.2019.2949587.
45. Zheng, X.; Niu, Y.; Chen, J.; Chen, Y. Color correction for stereoscopic image based on matching and optimization. 2017 International Conference on 3D Immersion (IC3D), 2017, pp. 1–8. doi:10.1109/IC3D.2017.8251900. 770
46. Sturm, P., Pinhole Camera Model. In *Computer Vision: A Reference Guide*; Ikeuchi, K., Ed.; Springer US: Boston, MA, 2014; pp. 610–613. doi:10.1007/978-0-387-31439-6\_472.
47. Rusinkiewicz, S.; Levoy, M. QSplat: A Multiresolution Point Rendering System for Large Meshes. Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques; ACM Press/Addison-Wesley Publishing Co.: USA, 2000; SIGGRAPH '00, p. 343–352. doi:10.1145/344779.344940. 775
48. Platt, J.; Others. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers* **1999**, *10*, 61–74.
49. Sharma, G.; Wu, W.; Dalal, E.N. The CIEDE2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations. *Color Research and Application* **2005**, *30*, 21–30. doi:10.1002/col.20070.