Computers & Graphics 106 (2022) 174–186

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Special Section on CEIG 2022

Gain compensation across LIDAR scans

Imanol Munoz-Pandiella [a,*], Marc Comino Trinidad [b], Carlos Andújar [c], Oscar Argudo [c], Carles Bosch [d], Antonio Chica [c], Beatriz Martínez [c]

[a] Universitat de Barcelona, Spain
[b] Universidad Rey Juan Carlos, Spain
[c] Universitat Politècnica de Catalunya, Spain
[d] Universitat de Vic - Universitat Central de Catalunya, Spain

A R T I C L E   I N F O

Article history:
Received 12 May 2022
Accepted 3 June 2022
Available online 18 June 2022

Keywords:
Gain compensation
LIDAR
Panorama
Color constancy
3D reconstruction

A B S T R A C T

High-end Terrestrial Lidar Scanners are often equipped with RGB cameras that are used to colorize the point samples. Some of these scanners produce panoramic HDR images by encompassing the information of multiple pictures with different exposures. Unfortunately, exported RGB color values are not in an absolute color space, and thus point samples with similar reflectivity values might exhibit strong color differences depending on the scan the sample comes from. These color differences produce severe visual artifacts if, as usual, multiple point clouds colorized independently are combined into a single point cloud. In this paper we propose an automatic algorithm to minimize color differences among a collection of registered scans. The basic idea is to find correspondences between pairs of scans, i.e. surface patches that have been captured by both scans. If the patches meet certain requirements, their colors should match in both scans. We build a graph from such pair-wise correspondences, and solve for the gain compensation factors that better uniformize color across scans. The resulting panoramas can be used to colorize the point clouds consistently. We discuss the characterization of good candidate matches, and how to find such correspondences directly on the panorama images instead of in 3D space. We have tested this approach to uniformize color across scans acquired with a Leica RTC360 scanner, with very good results.

1. Introduction

Lidar scanning is a robust 3D digitization technique extensively used in architecture, engineering, construction and cultural heritage. Complex scenes such as buildings, monuments and sites, must be scanned from multiple locations to get a sufficient coverage (Fig. 1). Terrestrial Lidar equipment is often mounted on a rotating support so as to deliver 360 data around the scanner location. The basic information acquired for each scan is a collection of point samples, each point having spatial (x, y, z) coordinates and an intensity value corresponding to the return of the infrared laser used for depth measurements. In order to acquire color information, state-of-the-art scanners are equipped with multiple cameras. For example, Leica's RTC360 scanner (Fig. 1) includes 3 cameras that take pictures at 12 horizontal orientations (30° shift). The scanner is able to acquire High Dynamic Range (HDR) images by encompassing the information of different exposures. The scanner captures 432 MP (12 MP × 3 cameras × 12 positions) at 5 HDR brackets [1]. After considering overlaps, the scanner is able to create 200 MP (20480 × 10240) panoramas (Fig. 2).

Similarly to current digital cameras, the scanner's software automatically applies image enhancement techniques to improve the color reproduction. For example, the RTC360 applies color balance and exposure correction to each scan before it is exported. The goal is to compensate the images as if illuminated with CIE standard illuminant D65, which roughly corresponds to daylight. This helps reducing the impact of poor illumination and color differences due to the presence of light sources with different illumination profiles. However, these image enhancement algorithms operate according to statistical properties of the acquired images, which largely depend on the point of view and thus the surfaces visible from a specific scan location. As a result, although individual panoramas are (overall) well-balanced in terms of color and exposure, the same surface patch can appear with very different colors depending on the scan location (Fig. 3 left). This is not a problem if scans are inspected individually; however, if as usual, multiple scans are combined into a single point cloud, large color deviations among neighboring samples produce severe visual artifacts (Fig. 3 right).

* Corresponding author.
E-mail addresses: imanol.munoz@ub.edu (I. Munoz-Pandiella), marc.comino@urjc.es (M. Comino Trinidad), andujar@cs.upc.edu (C. Andújar), oargudo@cs.upc.edu (O. Argudo), carles.bosch@uvic.cat (C. Bosch), achica@cs.upc.edu (A. Chica), beatriz.martinez@citm.upc.edu (B. Martínez).

https://doi.org/10.1016/j.cag.2022.06.003
0097-8493/© 2022 Published by Elsevier Ltd.

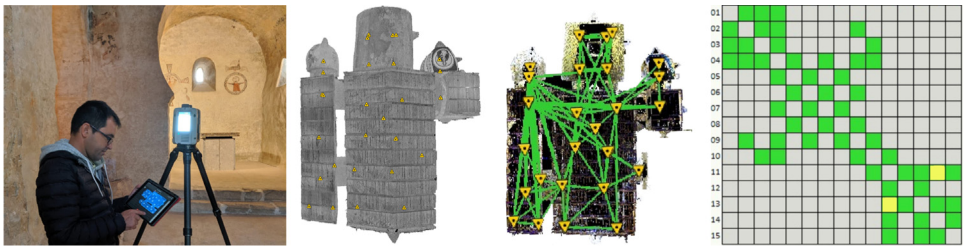

Fig. 1. The Leica RTC360 scanner (*left*) combines laser scanning and color cameras to capture colored point clouds. Multiple scans (*middle*) are required to digitize complex monuments such as this medieval church. Registration software often reports the quality of the links as a symmetric matrix (*right*).

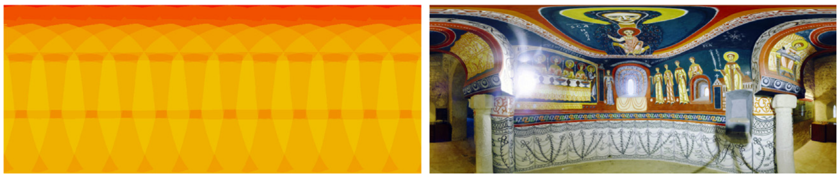

Fig. 2. Representation of the overlapping field of view of the RTC360 cameras (*left*), and example of HDR 360 panorama (*right*).

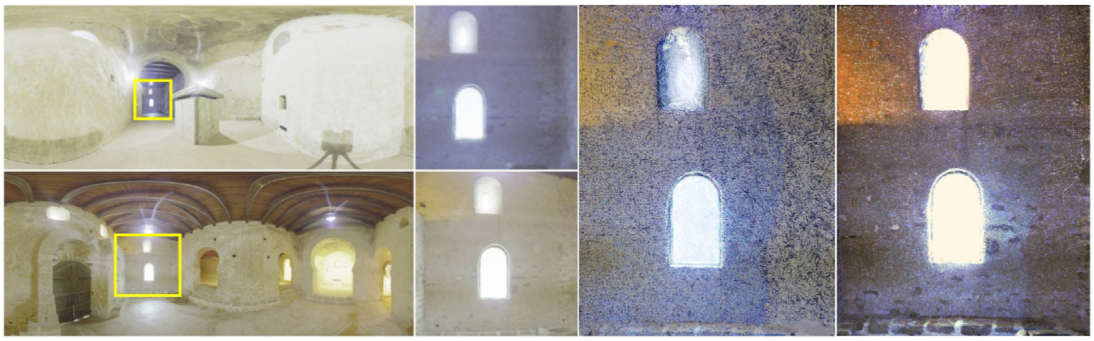

Fig. 3. Laser scanners apply automatic image enhancement techniques to individual panoramas (*left*). In challenging scenes with varied illumination profiles, the same surface (e.g. the highlighted wall) may appear with very different colors/brightness depending on the scan (*middle left*). Since these images are used to color the point samples, rendering simultaneously multiple registered scans (*middle right*) produces severe visual artifacts, with pseudorandom patterns according to the visibility of the points from the different scans. Using HDR images and advanced tone mapping techniques (*rightmost image*) does not solve the problem.

In this paper we propose an automatic algorithm to maximize color consistency *across panoramas*. We achieve this by computing RGB gain compensation factors for each panorama taking into account statistics of HDR color data (mean and median RGB color values). These factors somehow reverse the color enhancement corrections applied by the scanner. Doing so we (temporarily) sacrifice the color balance and exposure correctness of the individual scans, in favor of color coherence across them (Fig. 4). Once the RGB gain factors are applied to the panoramas, the color is transferred to the point cloud. We use HDR color values throughout the process. When needed, HDR panoramas are converted to LDR by applying a tone mapping operator (we used [2] for all the HDR panoramas shown in this paper). Notice that, after our correction, the point samples share coherent colors, and thus image enhancement and color balance techniques can be applied safely to the whole point cloud without producing artifacts.

We compute the compensation factors by detecting correspondences between pairs of scans, i.e. surface patches that are simultaneously visible by two scans. We use these pair-wise correspondences to define a graph that encodes the overlap relationships between panoramas. Terrestrial Lidar scans are usually acquired with some view overlap to facilitate registration, so we assume that the resulting graph is connected and thus that there is a path relating any panorama to any other panorama in the graph. Lidar registration software often report the links that have been used on-site to perform a rough pre-registration of the scans as well as a matrix representing the alignment quality of such links (Fig. 1) after fine registration. We compute a similar graph but based on a selection of simultaneously visible surface patches, and use it to find the best compensation factors.

Detecting matching elements across images is also an essential process in image stitching (reviewed in next section) and photogrammetry reconstruction. However, the problem we address is essentially different to image stitching in terms of the final use of the compensated images. Since we wish to use the compensated panoramas to colorize the Lidar point clouds, we cannot assume that neighboring pixels in image space correspond to neighboring surface samples in 3D space, as this assumption only makes sense for images taken from a single viewpoint, whereas in our case we deal explicitly with panoramas taken from distinct locations, since our goal is to uniformize colors across the whole Lidar dataset. In other words, adjacency between panorama pixels must be defined in terms of their corresponding (unprojected) 3D positions, and thus we use these positions to identify correspondences.

Another difference is the amount of information available that we exploit to find relevant correspondences between pairs of panoramas. As we shall see, we use the Lidar infrared values (besides normal and depth maps) to refine which pixels of the panoramas should be taken into account when computing the gain compensation factors.

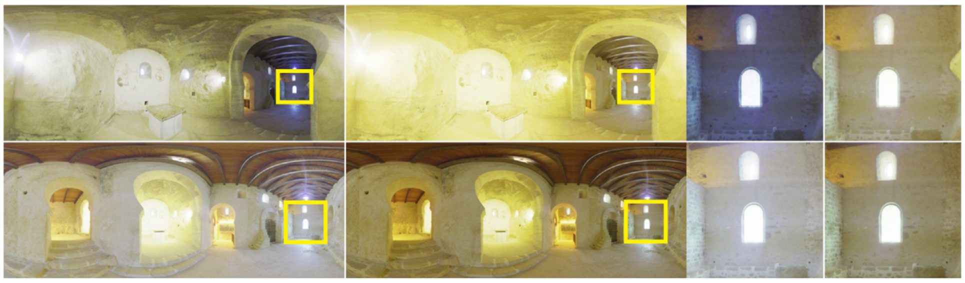

Fig. 4. Gain compensation example. The first column shows a pair of HDR panoramas, which have been uniformized with our approach (second column). Although we removed the effect of color balance (the central apse on the upper row had a strong yellowish illumination), we achieve a much more uniform color across scans. For example, the same wall appears with different brightness levels in the original panoramas (third column), which would lead to visual artifacts if the point clouds are rendered simultaneously. Our approach though achieves more uniform color (fourth column).

The rest of the paper is organized as follows. Section 2 reviews previous work on image stitching, HDR imaging and tone mapping. Section 3 presents our approach, and Section 4 presents our results with a large collection of Lidar scans. We conclude and discuss future work in Section 5.

2. Previous work

Image stitching Color balancing or correction plays a critical role in image stitching. It allows the composited image to show a natural and consistent color, and facilitates the blending step [3]. Initial efforts on solving the color balancing problem for multi-view stitching focused on gain compensation [4]. By adjusting the intensity gain level of images, they compensate for different exposure levels. Such compensation, can also be performed using all three color channels independently [5].

The broader concept of color correction arises from color transfer between images, which was first proposed by Reinhard et al. [6]. Color correction approaches can be divided into parametric and non-parametric ones. Parametric approaches assume that the mapping of color between images follows a known model, which has shown to produce some of the best results [5]. Most methods focus on correcting colors using a global solution, but local approaches based on matching specific regions are also available [7]. Non-parametric approaches, on the other hand, try to avoid building an explicit model by estimating a look-up table from relevant statistics of the images. Such approaches might be more flexible, but require enough variability in the shared regions to work appropriately [5]. More recent approaches put more emphasis on grain-free images, detail preservation and artifacts suppression [3].

HDR from multiple exposures In many applications of digital image capturing, processing and analysis, such as computer vision and 3D reconstruction, obtaining radiance and texture information for all the scene areas is critical. When dealing with scenes illuminated by a high range of light intensities, HDR imaging overcomes the camera sensors dynamic range limitations and enables the correct reproduction of all the luminosity characteristics of a scene. This improves the performance of many image processing and analysis tasks, including feature point detection [8].

HDR imaging is based on the capture, for a given point of view, of different photographs with different exposures. For digital cameras that store images in 12–14 bits raw mode, 3 shots with exposure differences of 2 EVs are enough to obtain optimal results [9]. These LDR images are fused to obtain a single image by applying different algorithms of HDR radiance image reconstruction, which reflect the radiance of the real-life scene. The first HDR algorithms were developed in the late '90s, and some of them [10,11] are still used in HDR imaging processes.

One of the differences among the existing algorithms is whether the method assumes a linear [12,13] or a non-linear camera response function [10,14,15]. Those of the latter group can be applied for raw images, which can be considered linear.

HDR images use 32 bits IEEE floating point values to represent each color channel in order to fully record all the radiance data in a scene. The high number of bits makes impossible to view these image data on a normal LDR display. Therefore, the compression of the HDR content to LDR content is needed, by applying a Tone Mapping Operator (TMO) [16], when the result of the HDR imaging must be visualized.

Tone mapping Tone mapping is key when dealing with HDR images. This importance is demonstrated by the large amount of techniques that can be found in the literature, in books [15,17, 18], as well as surveys [19,20]. TMOs compress the illuminance range while preserving contrast by using a mapping function. This may be done globally [16,21], applying the same function to all pixels, or locally by changing the function for each pixel depending on its neighbors [2,22–26]. As reflectance is a fundamental descriptor of a scene, reducing the dynamic range of the illuminance while preserving the reflectance, compresses global contrasts while preserving local ones, or details.

Another possibility is to classify tone mapping operators according to their effect on global contrast, contour and texture detail loss [27]. Changes in color appearance may be corrected, but the relation of contrast changes to color appearance is non-linear and not easily explained by color perception models [28]. Nevertheless, both local and global operators may be approximated by relatively simple image operations [29]. Noise is also a concern, as many operators may be affected by it, thus introducing contrast distortion and ringing artifacts. Having a local noise-aware TMO that minimizes such effects [30] may be very relevant depending on the used capturing device.

More recently, research has been oriented to develop application specific tone mapping operators. The advent of HDR video came with its own set of challenges [19]. Their use for HDR image matching for computer vision applications [31] is also not trivial. There, the accurate detection and description of keypoints is fundamental, which requires an invariance to transformations over local neighborhoods, while allowing for accurate localization of any detected keypoint position. We may train such application specific operators using genetic programming [32], or a deep learning architecture [33]. Tone mapping may also be applied to HDR images taken during LiDAR scanning [34]. The intensity (infrared reflectivity) captured during the laser scan is used as a guide to improve the exposure of the photographs. This same intensity can be used to correct color in LiDAR scans [35]. Well-exposed areas are used to train a model, that is then used to predict the color of problematic areas.

3. Approach

3.1. Inputs

We focus on computing consistent color across a 3D point cloud which has been acquired by range-scanning a real scene from a set $Q \subset \mathbb{R}^3$ of different scanner locations. We assume that we have access to the independent raw scans $C_{\mathbf{q}}$, and that the sensor location $\mathbf{q} \in Q$ from which they were captured is available. Moreover, these scans may be in their own coordinate system and, in this case, we also assume we have access to a 4×4 affine matrix $\mathbf{T_q}$ that registers them to a global coordinate system.

At capturing time, the scanner placed at $\mathbf{q} \in Q$ will emit several laser beams that upon reaching the scene surface π will yield points $\mathbf{p}$ with reflected infrared intensity $i_{\mathbf{p}}$. After finishing the geometry acquisition step, the scanner proceeds to capture and stitch several photographs to produce low (LDR) and a high (HDR) 360° equirectangular panoramas ($L_{\mathbf{q}}$ and $H_{\mathbf{q}}$, respectively). Usually the LDR image is computed by tone mapping the HDR image. While infrared information is perfectly aligned with the captured geometry, color information may suffer color bleeding artifacts due to misalignment and color-geometry miss-registration artifacts caused by dynamic objects (e.g. people).

Although we could convert the equirectangular panoramas to alternative 360° representations with less area distortion (e.g. cube maps), in this paper we preserve the equirectangular projection as its pixel spacing closely matches that of the point cloud, since the scanner cameras have been designed such that the optical resolution matches that of point cloud resolution at the highest setting of 3 mm @10 m point spacing [1].

3.2. Panorama preprocessing

Our color correction approach consists of applying a global operator to the $H_{\mathbf{q}}$ panoramas whose parameters are estimated from surface patches that are visible from pairs of scan locations $\mathbf{q} \in Q$. The first step of our method consists of finding these patches and, for this purpose, we follow Comino et al. [36] to compute depth and normal information for the surface represented by each pixel $H_{\mathbf{q}}(x, y)$. These quantities are estimated by ray-casting the point cloud and combining the information of multiple points using different splatting weights. These weights have been smartly crafted to ensure the sharpness of the result. The process yields the depth maps $D_{\mathbf{q}}$ and normal maps $N_{\mathbf{q}}$.

3.3. Flow maps

Next, we compute flow maps $F_{\mathbf{q},\mathbf{q}'}$ between a subset of the pairs of scan locations $\{(\mathbf{q}, \mathbf{q}')|\mathbf{q} \neq \mathbf{q}', \mathbf{q} \in Q, \mathbf{q}' \in Q\}$. A flow map $F_{\mathbf{q},\mathbf{q}'}$ at pixel coordinates (x, y) contains a triplet of values (m, x', y'). The value m is a binary indicator; if set to 1 then pixels $H_{\mathbf{q}}(x, y)$ and $H_{\mathbf{q}'}(x', y')$ see the same surface. Otherwise, it indicates that we could not find any pixel in $H_{\mathbf{q}'}(x', y')$ that represents the same surface as $H_{\mathbf{q}}(x, y)$.

To estimate the flow map $F_{\mathbf{q},\mathbf{q}'}$ we traverse each pixel (x, y) of the depth map $D_{\mathbf{q}}$ and perform the following steps:

1. We use the inverse equirectangular projection to obtain the 3D point $\mathbf{p}$ corresponding to $(x, y, D_{\mathbf{q}}(x, y))$.
2. Next, we convert this point to the coordinate system of the scan $C_{\mathbf{q}'}$ by performing $\mathbf{p}' = (\mathbf{T_{q'}})^{-1}\mathbf{T_q}\ \mathbf{p}$.
3. Then, we project $\mathbf{p}'$ using the equirectangular projection to obtain the pixel coordinates (x', y') and reprojected depth d'_{rp}.
4. Since (x', y') are real coordinates, we perform bilinear interpolation on $D_{\mathbf{q}'}$ to obtain the corresponding depth d'.

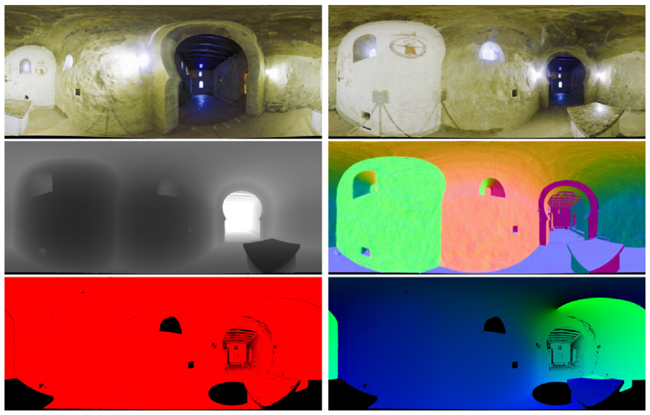

Fig. 5. From top to bottom and left to right: one panorama $H_{\mathbf{q}'}$, a second panorama $H_{\mathbf{q}}$, its depth map $D_{\mathbf{q}}$, its normal map $N_{\mathbf{q}}$, their overlap as seen from $\mathbf{q}$ (red channel of the flow map $F_{\mathbf{q},\mathbf{q}'}$), and the flow map itself $F_{\mathbf{q},\mathbf{q}'}$ (green and blue channels).

5. If the difference between the reprojection depth (d'_{rp}) and the depth captured from $\mathbf{q}'$ (d') is smaller than a certain threshold (we use 1 cm in our examples) then we assume pixels $D_{\mathbf{q}}(x, y)$ and $D_{\mathbf{q}'}(x', y')$ represent the same surface and set $F_{\mathbf{q},\mathbf{q}'}(x, y) = (1, x', y')$.
6. Otherwise, we set $F_{\mathbf{q},\mathbf{q}'}(x, y) = (0, 0, 0)$.

Fig. 5 shows an example of flow map, where red (overlap regions) and green/blue channels (actual pixel coordinates) have been separated for clarity.

There are several criteria for choosing the scan pairs $(\mathbf{q}, \mathbf{q}')$ for which we compute their flow maps $F_{\mathbf{q},\mathbf{q}'}$, since considering all pairs has weighted cost. One possibility is to use the links from the alignment quality matrix, if available (Fig. 1), since these links guarantee a certain overlap. Another option is to start from a central scan $\mathbf{q}$ (near the centroid of Q), compute all flow maps from $\mathbf{q}$, and proceed similarly from the related scans until we get a connected graph that includes all scans in Q. Flow maps can be generated at a smaller resolution than actual HDR images, since we will use them to compute three RGB compensation factors for each panorama.

3.4. Refining pair-wise correspondences

Although flow maps already provide an initial correspondence between scans, they can include regions with undesirable properties: as we aim to correct the color of surfaces visible from several locations, any surface property that affects surface radiance and depends on the viewpoint will lead to computing wrong compensation factors. For this reason, we propose to first divide each flow map $F_{\mathbf{q},\mathbf{q}'}$ in several patches and, then, filter these patches to only take into account the parts where these undesirable properties do not occur.

We divide each flow map $F_{\mathbf{q},\mathbf{q}'}$ by computing superpixels on it using MSLIC algorithm [37], where each superpixel corresponds to a patch. Using flow maps information to compute patches is suitable as it encodes the correspondence of the visibility of $\mathbf{q}'$ in $\mathbf{q}$, thus avoiding handling nonvisible surfaces. Moreover, in most situations, discontinuities on the data of the flow map $F_{\mathbf{q},\mathbf{q}'}$, which determine superpixels' border, represent changes on the captured surface. Furthermore, the size of the computed superpixel is also important. Each patch must be large enough to allow the computation of reliable statistics on the associated color data (see $c_{\mathbf{q}}$ in Section 3.6), but, at the same time, it must be small enough to keep low the likelihood of the patch having several undesired properties. In our experiments, we have found that an average superpixel size of 50 pixels satisfies both conditions.

After that, we filter each patch discarding those that exhibit undesired characteristics. For each patch, we compute a score ($sc_i \in [0, 1]$) that evaluates the following properties, where 0 means discarding the patch and 1 considering it.

Tangent surfaces Patches with normals not oriented towards the sensor correspond to surfaces that are poorly captured by the scanner. We compute the average normal of each patch regarding the view direction of the sensor by sampling the normal map $N_{\mathbf{q}}$ at each pixel of the superpixel. We accept ($sct_i = 1$) those patches whose average normal has an angle $\alpha \leq 15^\circ$ with respect to the direction towards the sensor; we discard ($sct_i = 0$) those with $\alpha > 70^\circ$, and we linearly penalize those between these angles.

Rough regions A rough surface might show faces with different orientations depending on the viewpoint. So, the set of faces visible from $\mathbf{q}$ might get different irradiance than those visible from $\mathbf{q}'$. For each pixel of the superpixel, we sample the normal map $N_{\mathbf{q}}$ and compute the standard deviation of the normals $sdt(n_i)$ inside the patch. We compute the score of this property as the relation between the standard deviation of the superpixel with respect to the maximum of the panorama: $scr_i = \frac{sdt(n_i)}{\max(sdt(n_i))}$.

Specular materials Specular surfaces have view-dependent radiance and, thus, cannot be used for color uniformization. Since specular materials bounce back to the sensor low intensity values, for each patch, we compute the minimum infrared intensity $i_{\mathbf{p}}$ of the superpixel. We accept ($scs_i = 1$) those patches that $\min(i_{\mathbf{p}}) > 0.15$, we discard those that $\min(i_{\mathbf{p}}) \leq 0.07$, and we linearly penalize those between that infrared intensity values.

Dark parts As we need to retrieve statistics from R, G, and B channels, we need to use bright patches: a too dark one would provide unreliable data. To avoid them, we compute the lightness of a patch by sampling the LDR image $L_{\mathbf{q}}$. For each pixel, we convert its RGB color to HSV and compute the median lightness value of the superpixel V_i. We linearly penalize superpixels depending on this lightness median value: $scd_i = V_i$.

Zones dispersed in $\mathbf{q}'$ Large disparity between the size of a patch in $\mathbf{q}$ and $\mathbf{q}'$ would bring us to retrieve statistics of non-comparable regions. To avoid that, we compute the gradient of the flow $F_{\mathbf{q},\mathbf{q}'}$ using Sobel operators with a 3×3px kernel. Let $Sx_{\mathbf{q},\mathbf{q}'}$ be the horizontal component (green channel) of $F_{\mathbf{q},\mathbf{q}'}$ and $Sy_{\mathbf{q},\mathbf{q}'}$ the vertical one (blue channel). For each pixel, we compute the disparity as the maximum value of the gradient $DS_{\mathbf{q},\mathbf{q}'} = \max(\partial Sx_{\mathbf{q},\mathbf{q}'} \partial x, \partial Sy_{\mathbf{q},\mathbf{q}'} \partial y)$. For each superpixel, we compute the median value of this disparity $\tilde{ds}$. We accept ($scf_i = 1$) those patches that their median disparity $\tilde{ds} \leq 10$ pixels, we discard ($scf_i = 0$) those that $\tilde{ds} > 15$ pixels, and we linearly penalize between those values.

The final score of a patch is computed by combining those obtained for each property:

$$sc_i = sct_i \cdot scr_i \cdot scs_i \cdot scd_i \cdot scf_i$$

As we will show in Section 3.6, this simple combination allows the user to effortlessly tune the filtering results using an easy-to-understand threshold. This threshold will define which patches are not suitable to estimate the corresponding RGB gain factors and, thus, should be discarded.

3.5. Building the graph

As we compute pairwise correspondences, we compute a graph $\mathcal{G}$ whose nodes correspond to scan origins $\mathbf{q} \in Q$ and edges represent valid overlaps between pairs of panoramas (Fig. 6), each overlap consisting of at least one patch. In a continuous setting, the part of a scene that is simultaneously visible from $\mathbf{q}$ and $\mathbf{q}'$ is unequivocally defined by $\mathbf{q}$ and $\mathbf{q}'$. In a discrete setting,

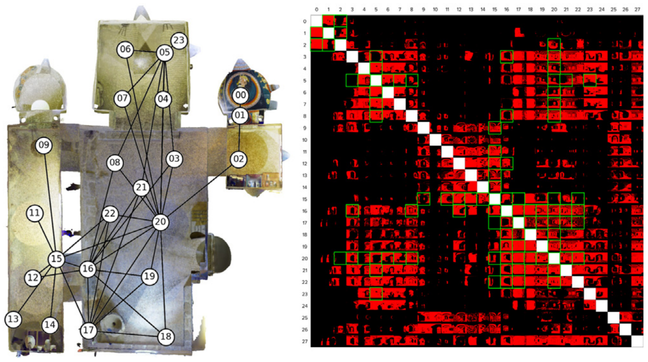
Fig. 6. Undirected graph encoding valid correspondences between panoramas (*left*) and flow maps (only red channel) showing matching pixels across panorama pairs (*right*). We have omitted some scan locations very close to others shown in the graph. For instance, scan 24 and 27 were omitted because they are very close to scans 6 and 20, respectively.

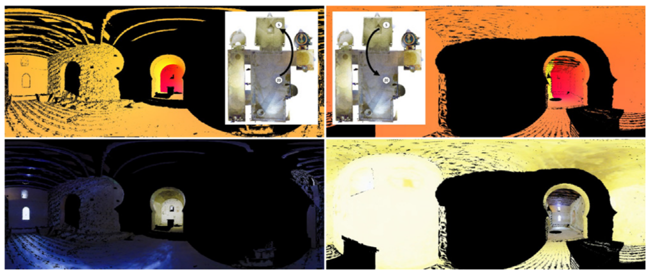
Fig. 7. Symmetric flow maps differ due to discretization but roughly represent the same set of 3D points/pixels of the panoramas. The top row shows $F_{\mathbf{q}_{20},\mathbf{q}_{05}}$ and $F_{\mathbf{q}_{05},\mathbf{q}_{20}}$. The inset images indicate the location of $\mathbf{q}_{20}$ and $\mathbf{q}_{05}$ on an overhead map. The bottom row shows the color warped from one panorama to the other using the flow maps, just for illustration purposes.

Fig. 8. One of the panoramas of the test scene. Notice the presence of multiple types of light sources.

a flow map $F_{\mathbf{q},\mathbf{q}'}$ and its reverse $F_{\mathbf{q}',\mathbf{q}}$ do not represent exactly the same subset of 3D points, due to the spatial discretization of the flow maps, and the operations described above to refine correspondences. In practice, the differences are minimal in terms of the statistical properties (see $c_{\mathbf{q}}$ in Section 3.6) we use to compute the gain compensation factors (Fig. 7), and thus we consider $\mathcal{G}$ to be an undirected graph.

3.6. Solving for RGB gain factors

For each RGB color channel, our goal is to compute a set of factors $\{\lambda_{\mathbf{q}}\}$ for each $H_{\mathbf{q}}$, $\mathbf{q} \in Q$ to compensate the intensity gain. We propose two different approaches to compute this set.

The first one is using a **linear system**. For every edge in $\mathcal{G}$ between panoramas $H_{\mathbf{q}}$ and $H_{\mathbf{q}'}$, we define an equation: $c_{\mathbf{q}}\lambda_{\mathbf{q}} = c_{\mathbf{q}'}\lambda_{\mathbf{q}'}$, where the constant $c_{\mathbf{q}}$ is a measured statistic about the

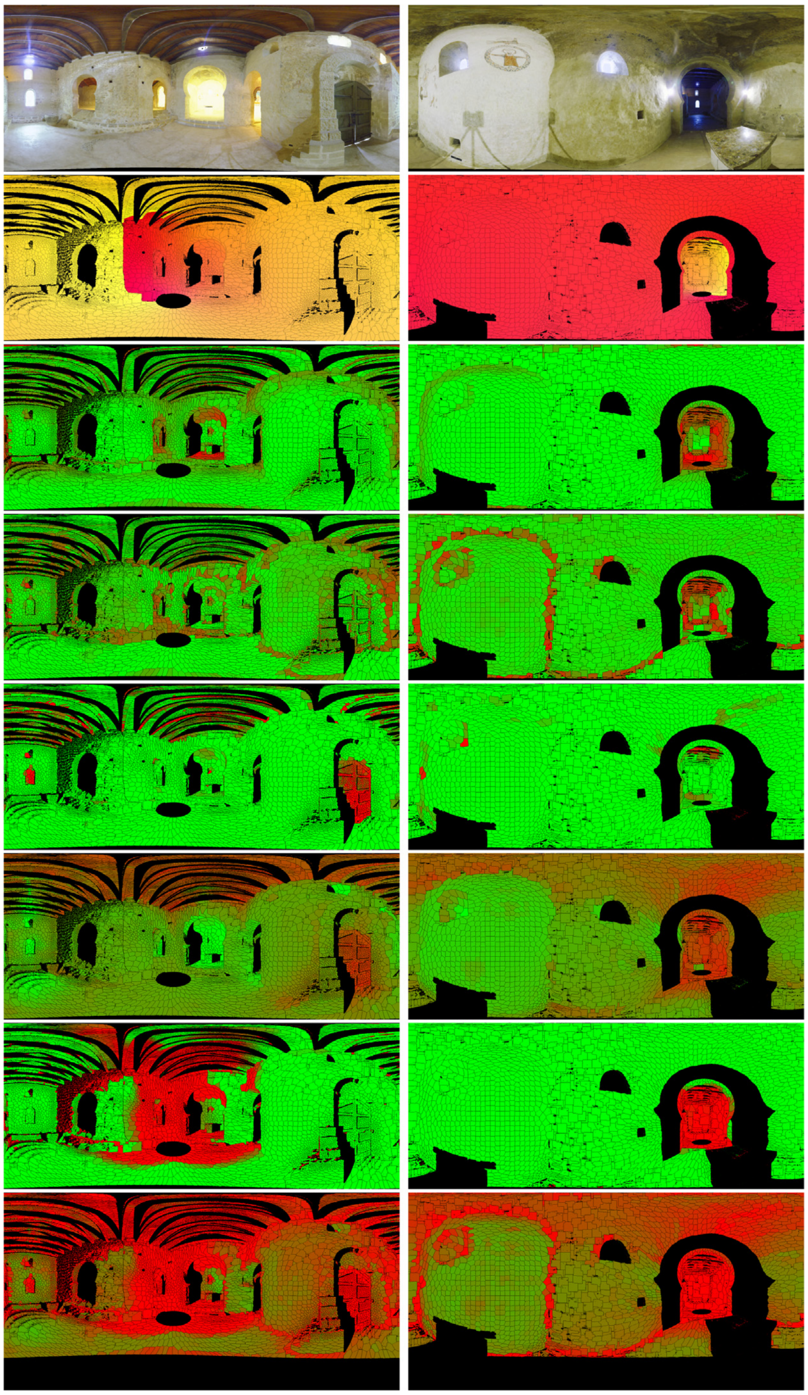

Fig. 9. Result of applying the different filters (and their combination) to panoramas 20 (left) and 5 (right). From top to bottom: $H_{\mathbf{q}}$, superpixels computed on $F_{\mathbf{q},\mathbf{q}'}$, tangent surfaces filter (sct_i), rough regions filter (scr_i), specular materials filter (scs_i), dark parts filter (scd_i), dispersed zones filter (scf_i) and their combination (sc_i) using 0.1 as threshold. Superpixels of the last row are colorized according to their sc_i factor (red for $sc_i = 0$ and green for $sc_i = 1$). They represent valid and invalid correspondences for the compensation factors above.

intensity in the pixels from $H_{\mathbf{q}}$ that overlap with $H_{\mathbf{q}'}$, and respectively $c_{\mathbf{q}'}$. We have experimented using the mean intensity and the median intensity. This system is largely overdetermined: it has $n = |Q|$ unknowns and $m = O(|Q|^2)$ equations, since the number of flow maps can be quadratic and we can sample more than one overlapping patch per flow map. Therefore, we use least squares to find the approximate solution that minimizes the norm:

$$\min_{\boldsymbol{\lambda}} \|(\mathbf{C_q} - \mathbf{C_{q'}})\boldsymbol{\lambda}\|$$

where $\boldsymbol{\lambda}$ is the column vector with the n factors and $\mathbf{C_q}$ is an $m \times n$ matrix with only one value per row: c_q in the column corresponding to the index of panorama q.

Note that this system has the trivial solution $\boldsymbol{\lambda} = 0$. Thus, we choose one panorama $\mathbf{q}_i$ as the reference, set its factor to $\lambda_{\mathbf{q}_i} = 1$, and modify the equations accordingly.

In our second approach, we minimize a **quadratic cost** representing the weighted sum of squared differences:

$$\min_{\boldsymbol{\lambda}} \sum_{i=0}^{m} \omega_i (c_{\mathbf{q}}\lambda_{\mathbf{q}} - c_{\mathbf{q}'}\lambda_{\mathbf{q}'})^2$$

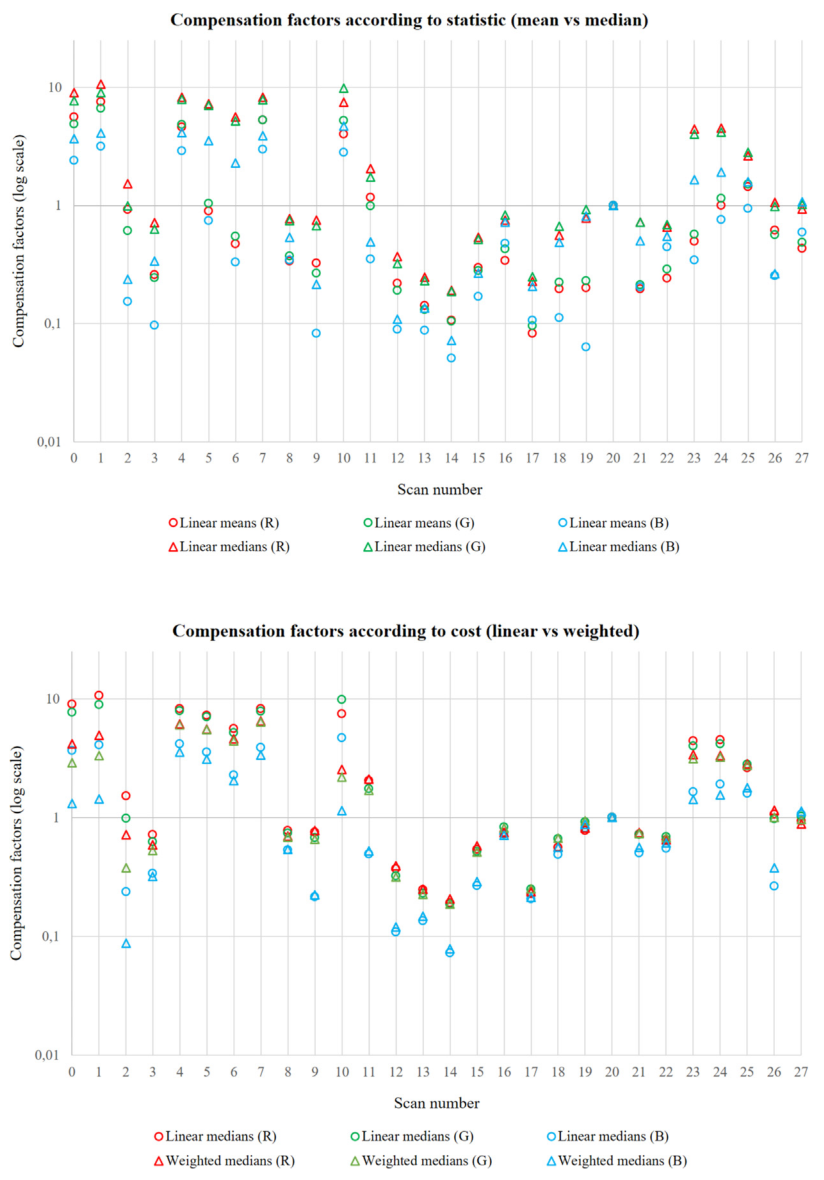

Fig. 10. Per-component gain factors computed for each panorama using panorama 20 as reference ($\lambda_{20} = 1$ for all three color channels).

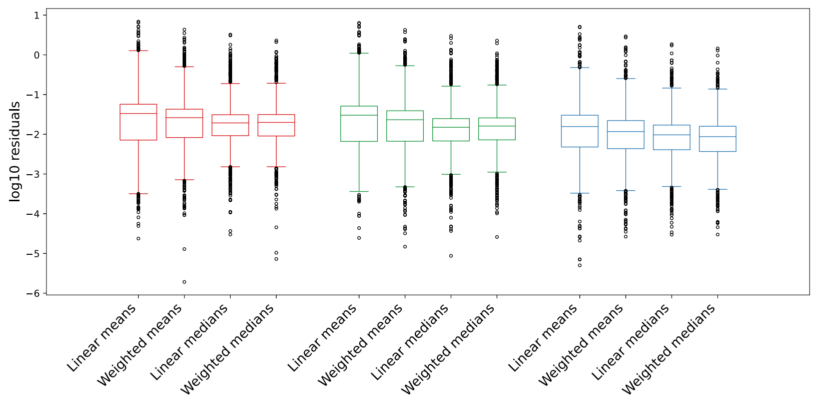

Fig. 11. Box plot of patch residuals after gain compensation, grouped by color component (R,G,B).

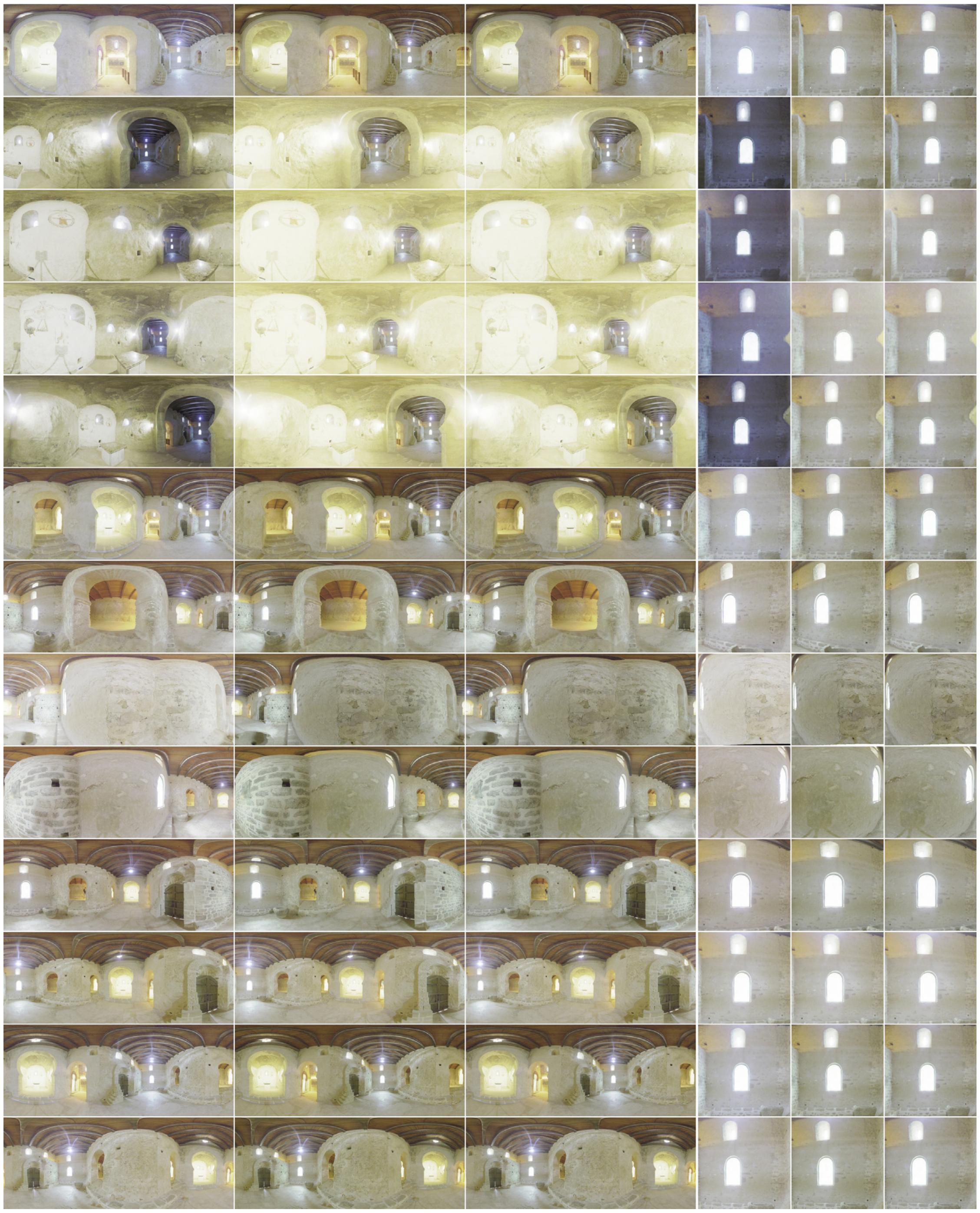

Fig. 12. Results, for a complete panorama (left group) and zoom in (right group), after compensating the original panorama (*first column*) with the linear system (*second column*) and the weighted cost solution (*right column*), operating on **RGB medians**. The zoom in shows the same wall in both original (*left column*) and compensated images (*second and third column*). We only show a subset of the panoramas capturing the wall.

We defined the weights as $\omega_i = sc_i \cdot \frac{\min(\text{size}_q, \text{size}_{q'})}{\text{panorama size}}$, where sc_i is the final score of the patch, and the second term is the minimum of the relative sizes of the patch in either panorama.

Note that, in these cost functions, we only consider those patches with a score larger than a given user-defined threshold. We empirically found that $sc_i > 0.1$ provides the best quality since it includes a significant number of patches and achieves coherent intensities over all panoramas.

4. Results

Source code is available at https://gitrepos.virvig.eu/research/panorama-gain-compensation.

4.1. Test dataset

We have tested our approach with a collection of +20 scans of a medieval Church (Sant Quirze de Pedret, Fig. 1). We used a Leica

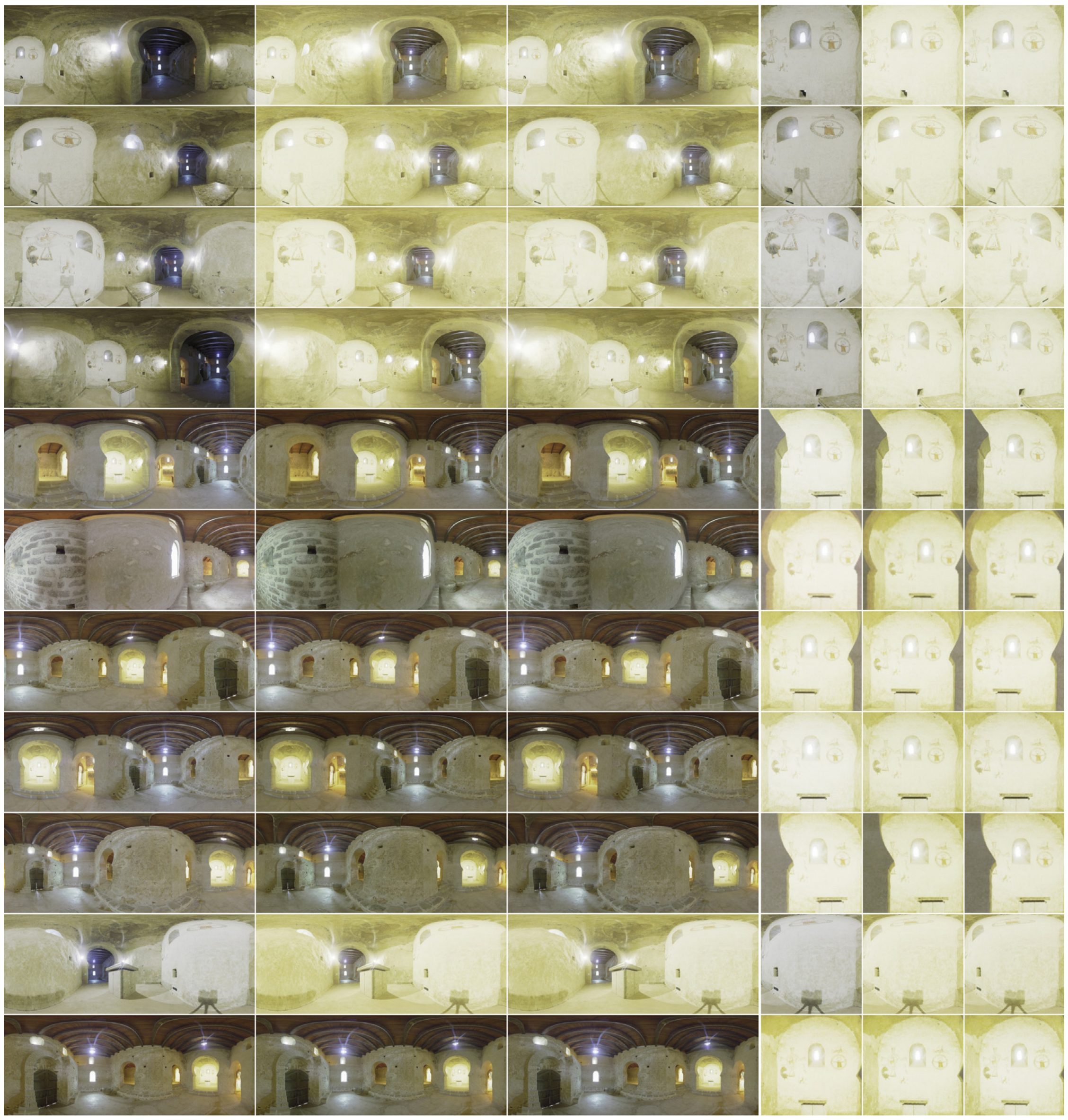

Fig. 13. Results, for a complete panorama (left group) and zoom in (right group), after compensating the original panorama (*first column*) with the linear system (*second column*) and the weighted cost solution (*right column*), operating on **RGB medians**. The zoom in shows the same wall in both original (*left column*) and compensated images (*second and third column*). We only show a subset of the panoramas capturing the wall.

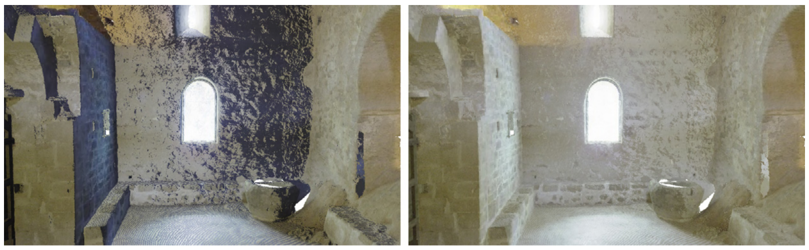

Fig. 14. Joint visualization of the point clouds acquired from scans 7 and 20 before (left) and after (right) applying our technique to the corresponding panoramas.

RTC360 laser scanner. The scene is quite challenging because it contains a variety of light sources with different profiles (Fig. 8): different types of fluorescent tubes, tungsten bulbs and led lamps, as well as daylight entering through multiple windows. Yellowish illumination was dominant in the apses due to lamps illuminating the paintings, whereas the central nave was dominated by daylight. As a consequence, any global color balance operator will fail to obtain reasonable results for the whole image, and local color balance operators operating on image space would emphasize color inconsistencies across scans.

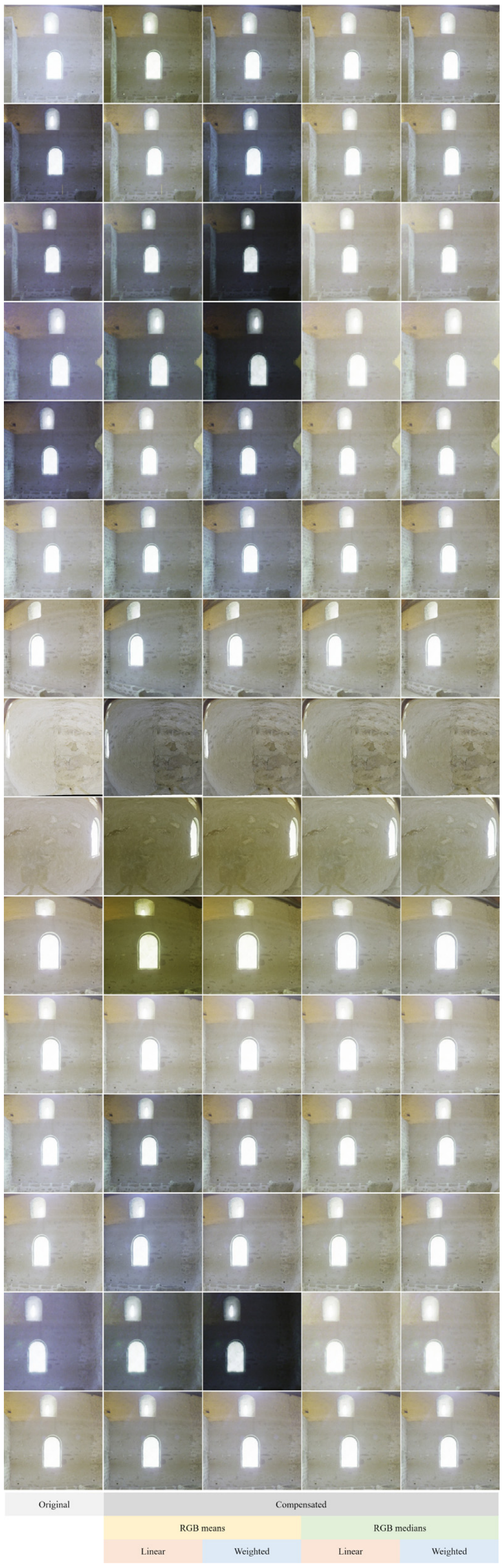

Fig. 15. Ablation study for the wall.

4.2. Flow maps

As a per-panorama preprocessing step, we computed all depth, normal and IR maps at a reduced (2048 $\times$ 1024) resolution. We computed these maps processing an average of 40,125 points/s using a CPU-based C++ implementation running on a desktop computer with an Intel Core i7-10700K CPU, 32 GB of RAM and a Kioxia Exceria Plus 500 GB SSD.

Then we computed a collection of pair-wise flow maps until we got a connected graph. We generated all flow maps starting from a few well-connected panoramas. Fig. 6 shows the resulting flow maps (only red channel, i.e. matching pixels across panorama pairs). In our experiments, we computed Flow maps of 2048 $\times$ 1024 pixels using depth maps of the same resolution as input. Each flow map took an average of 39.31 s to compute using a CPU-based python implementation running on the same computer.

4.3. Pair-wise correspondences

Fig. 9 shows the result of applying the pair-wise correspondences filters to different panoramas $H_{\mathbf{q}}$ (first row). The superpixels computed on the flow maps (second row) divide each panorama into patches of desirable size taking into account surface features and visibility between panoramas. Moreover, the figure shows how the different filters applied to each patch allow us to avoid regions with undesirable properties. The tangent surfaces filter (sct_i) discards poorly captured surfaces while preserving those with a higher sampling rate as flat perpendicular walls and spherical roofs of the absis. The rough regions filter (scr_i) prevents us to consider rough elements like rocks and bumpy walls. The specular materials filter (scs_i) rejects specular elements such as windows, well-polished elements (e.g. top of the communion table), and most of the varnished wood (e.g. beams and doors). The dark parts filter (scd_i) consistently analyzes the luminance of the scene and helps us to focus on the bright zones. The dispersed zones filter (scf_i) shows to be a good option to discard regions captured at very different distances between scans (e.g. floor near the scanning location $\mathbf{q}$). Finally, the combination of all the filters (sc_i) using 0.1 as the threshold value , integrates all the previous filters helping us to work only on regions with the desired properties and thus providing more reliable statistical color information.

In our experiments, we computed these filters for each flow map $F_{\mathbf{q},\mathbf{q}'}$ using an unoptimized single-core implementation using OpenCV and NumPy. The computation of the superpixels took 23 s, the tangent filter (sct_i) 6 s, the roughness filter (scr_i) 9 s, the specular filter (scs_i) 10 s, the dark parts filter (scd_i) 10 s and the dispersed zones filter (scf_i) 28 s. As several computations are shared between filters, the computation took 66 s altogether.

4.4. Gain factors

To compute the gain factors, we manually set panorama 20 as the reference scale, i.e. $\lambda_{20} = 1$. Overall, 605,914 patches between flow maps were used to define either the linear equations or the weighted quadratic cost function. For the later, we defined the weight of a patch as its size relative to the full panorama. Each color channel was solved independently. Fig. 10 shows the values obtained depending on the statistic used (mean/median) and the cost function (linear/weighted). For all scans, the gain factor applied to the blue channel is lower than the values applied to red and green channels. These results imply a general color correction to a yellowish cast, more or less intense among scans, as the predominant light source in each of them depends on the scanner point of view. For example, the values obtained for scan 12 indicate a remarkable change to a darker and warmer image, while scan 22 only suffers a subtle correction for color and exposure, when the gain factors are calculated with weighted medians solution. Regarding the results obtained with mean/median

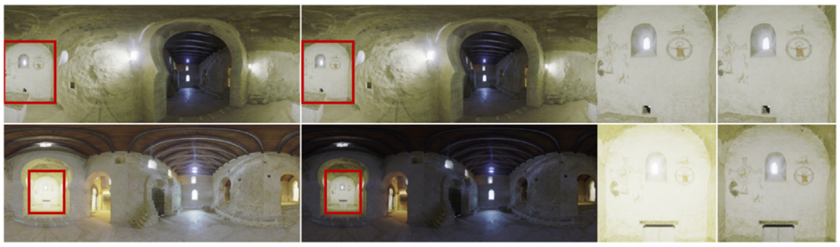

Fig. 16. White balance correction example (original panoramas and compensated ones) focusing on the central apse. Gain factors were optimized taking as reference scan 20 (outside the apse) but here we applied them taking as reference scan 04 (inside the apse), i.e. we applied gain factors $\lambda^c_{\mathbf{q}_i}/\lambda^c_{\mathbf{q}_{04}}$ instead of $\lambda^c_{\mathbf{q}_i}$. The white balance improvement can be seen by comparing the last column to that in Fig. 13.

Fig. 17. Scans taken under significantly different illumination conditions (main door open) would influence the results.

statistics, in many scans the values calculated with means are lower than those calculated with medians, for the three RGB channels. These lower values involve darker images, and in some cases the difference is highly relevant, also affecting chromaticity, as in scans 5, 6 and 19. On the other hand, the differences between linear and weighted solutions are hardly noticeable. Only scans 0, 1, 2 and 10 show differences between the two methods, affecting the exposure more than chromaticity.

To validate the obtained factors, we measure the absolute difference in each patch pair after applying the factors to the corresponding panoramas, i.e. the residual $|c_{\mathbf{q}}\lambda_{\mathbf{q}} - c_{\mathbf{q}'}\lambda_{\mathbf{q}'}|$. Only the patches larger than 900 pixels were considered to account for their larger visual impact in the final corrected panoramas. Fig. 11 shows the boxplot of the residuals, with whiskers equal to 1.5 the Interquartile Range. Although not statistically different, the median statistic produces lower residuals with a narrower range, and weighted cost yields slightly better results for means.

Regarding the timings, building the system of equations took about 0.4 s in our computer, plus 0.5 s to obtain the factors using Numpy linear least-squares solver. Since we process each color component individually, the total time to compute the factors was around 3 s.

4.5. Visual evaluation

Figs. 12 and 13 show our results using per-patch **RGB medians**. Fig. 12 compares the color coherence in the wall located in front of the central apse of the church. Despite the wall appears from a different view and sampling rate in each of the panoramas, it allow us to evaluate visually the color consistency across scans. Notice that the wall appears at a variety of exposures and color temperatures in the original panoramas (first column in each block). For example, the panoramas taken inside the central apse (second to fifth rows) exhibit a bright, warm lighting. The scanner adjusted the color balance and exposure for the central apse, resulting in a dark and blueish wall in the central nave. Rendering the combined point cloud with these colors results in severe artifacts (Fig. 3). Our approach succeeds in getting highly coherent color across scans (second and third column in each of the blocks), the weighted cost solution providing slightly better visual results than the linear system.

Fig. 13 compares the color coherence in the central apse. Again, the apse appears from different view angles and very different sampling rates. As in the previous example, the apse appears with different color temperatures in the original panoramas (first column in each block), depending on whether the scan was taken from inside the apse (the scanner's color balance achieved a gray appearance) or from outside. Our approach succeeds in getting highly coherent color across scans (second and third column in each of the blocks), with hardly noticeable differences between the linear and weighted solutions. Once all panoramas have been corrected, we could safely apply global or local color balance operators to the point cloud.

Fig. 14 shows the point cloud resulting from combining the acquired points for two panoramas (scans 7 and 20). Each point cloud is colorized using the corresponding panorama before (left) and after (right) applying our gain compensation technique. Despite some slight differences between panoramas that are still visible (mainly due to white balance and color calibration), the improvement of the resulting image is significant. The computed compensation factors minimize the differences of color between points, reducing the artifacts and leading to a more homogeneous result.

4.6. Ablation study

We now evaluate the benefits of using RGB means vs RGB medians, and linear vs weighted cost solutions. Fig. 15 shows the result of focusing on the wall in front of the central apse. All methods achieve better color coherence with respect to the original panoramas. When using RGB means, some scans are not adequately compensated (central rows). Our best explanation is that RGB mean values are very sensitive to outlier values in the HDR images, which by definition represent a broad range of radiance values, affecting the gain compensation factors e.g. for panoramas taken from locations capturing extreme outliers (e.g. direct sunlight). Actually, we observed that our HDR panoramas included values two orders of magnitude larger than the standard deviation. RGB medians are robust against outliers and achieve a better uniformization, with little differences between the linear and weighted solutions.

4.7. Color balance

Since applying the gain factors to the color channels of the panoramas just involves a multiplication, gain factors can be applied on-the-fly (e.g. in a fragment shader), leaving panoramas unchanged. Keeping the gain factors separated from the panoramas they affect allows us to quickly readjust them so as to take as reference a different panorama, therefore globally changing

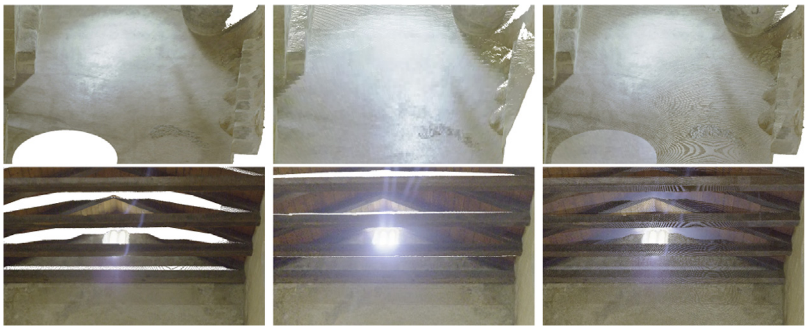

Fig. 18. Although our technique is able to correctly estimate gain factors in panoramas, view-dependent lighting effects such as specular reflections (*top*) and lens flares (*bottom*) will still produce artifacts when combining several scans into a single point cloud (right). Scan 20 is used on the *left*, scan 7 on *top center* and scan 19 in *bottom center*.

white balance. Let $\lambda^c_{\mathbf{q}_i}$ and $\lambda^c_{\mathbf{q}_j}$ be the gain factors computed for panoramas $\mathbf{q}_i$ and $\mathbf{q}_j$ for one of the RGB color channels c. Taking panorama $\mathbf{q}_j$ as the new reference just involves applying to $\mathbf{q}_i$ the gain factor $\lambda^c_{\mathbf{q}_i}/\lambda^c_{\mathbf{q}_j}$. This adjustment might be desirable for challenging scenes with varying illumination profiles (Fig. 16). This opens the possibility to adjust dynamically the color balance of the panoramas at runtime, depending on the part of the model being inspected, while preserving color consistency across scans.

4.8. Memory consumption

To compute the gain factors, for each relation between two different scan locations, we use one albedo map (3 channels, 1 byte per channel), one intensity map (1 channel, 2 bytes per channel), one normal map in world-space (3 channels, 2 bytes per channel), one normal map in eye-space (3 channels, 2 bytes per channel), one flow map (2 channels, 2 bytes per channel) and two HDR images (3 channels, 4 bytes per channel). In our experiments, we use these maps at a resolution of 2048×1024 pixels that leads to a consumption of 66 MBytes. Note that this consumption is temporal and it is freed once the score sc_i and the statistic $c_{\mathbf{q}}$ are computed for each superpixel. Moreover, our technique does not require to store the corrected panoramas as the application of the gain factors is straightforward. Although Fig. 6 can suggest we need a quadratic number of flowmaps, it only requires a subset of the available scans. In our experiments we used about $2N$ couples (where N is the number of scans). Besides that, the memory consumption can be reduced working in a lower resolution, as we only estimate one gain factor per channel and panorama.

4.9. Limitations

Although we obviously support spatially varying illumination, our method is sensitive to time-varying illumination, i.e. panoramas where *the same surface* receives significantly different illumination depending on the time the scan was taken. A certain amount of time-varying illumination will always occur due to multiple factors, e.g. clouds and Sun position affecting daylight. Our scans were taken in a time span of about 3 h, and although some lighting changes can be observed in the central nave, the overall color consistency is good. However, large time-varying illumination would impact negatively the color uniformity achieved by our method. For example, we had to remove one scan that was taken with the main door open (Fig. 17) as it completely changed the radiance inside the church (the door opening is much larger than the multiple narrow windows). Lens flares also cause some artifacts (see e.g. the lens flare around the central window in Fig. 8) and prevent a more accurate color consistency in certain parts of the panoramas. The same applies to specular reflections (see floor in Fig. 8), which are not considered during the gain compensation and might still result in view-dependent inconsistencies. Fig. 18 shows that our technique is not able to correct these situations and how they affect to the final point cloud.

5. Conclusions

Color consistency among scans capturing the same surface is essential to prevent highly apparent visual artifacts when rendering the combined point cloud. The automatic image correction applied by the scanner tends to be consistent for scenes with uniform illumination (e.g. daylight outdoor scenes), but not for scenes with varied light temperatures, which are corrected depending on the dominant surfaces seen from each scanner location. This problem also occurs in conventional photography, but its incidence in panoramas is higher due to their 360° nature. In this paper, we have presented a completely automatic algorithm for compensating HDR panoramas so as to maximize color consistency across scans. Identifying surface correspondences between scan pairs in image space instead of 3D space results in simpler code and better performance. Recolorizing the point clouds with the consistent panoramas dramatically reduces visual artifacts in the combined point cloud. Since we keep HDR colors during the whole process, exposure, color balance and tone mapping operators can be safely applied afterwards, even in an adaptive manner while rendering. Further postprocessing operations (e.g. subsampling, simplification, meshing and texturing) would also benefit from our approach, as these operations usually combine colors using local information on a small neighborhood. As future work, we wish to explore how to handle time-varying illumination. One possibility is to exploit the intensity values, since the infrared beam is nearly insensitive to lighting [34]. We also plan to extend our approach to handle non-diffuse surfaces and lens flares by exploiting existing information from other panoramas.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work has been partially supported by the project TIN2017-88515-C2-1-R funded by MCIN/AEI/10.13039/5011000-11033/FEDER “A way to make Europe”, by the EU Horizon 2020, JPICH Conservation, Protection and Use initiative (JPICH-0127) and the Spanish Agencia Estatal de Invesigación (grant PCI2020-111979), by the Universidad Rey Juan Carlos through the Distinguished Researcher position INVESDIST-04 under the call from

17/12/2020, and a Maria Zambrano research fellowship at Universitat Politècnica de Catalunya funded by Ministerio de Universidades.

References

[1] Leica RTC360 image resolution. White paper. 9435 Heerbrugg, Switzerland: Leica Geosystems AG; 2019.
[2] Reinhard E, Devlin K. Dynamic range reduction inspired by photoreceptor physiology. IEEE Trans Vis Comput Graphics 2005;11(1):13–24.
[3] Xia M, Yao J, Xie R, Zhang M, Xiao J. Color consistency correction based on remapping optimization for image stitching. In: Proceedings of the IEEE international conference on computer vision workshops. 2017, p. 2977–84.
[4] Brown M, Lowe DG. Automatic panoramic image stitching using invariant features. Int J Comput Vis 2007;74(1):59–73.
[5] Xu W, Mulligan J. Performance evaluation of color correction approaches for automatic multi-view image and video stitching. In: 2010 IEEE computer society conference on computer vision and pattern recognition. IEEE; 2010, p. 263–70.
[6] Reinhard E, Adhikhmin M, Gooch B, Shirley P. Color transfer between images. IEEE Comput Graph Appl 2001;21(5):34–41.
[7] Tai YW, Jia J, Tang CK. Local color transfer via probabilistic segmentation by expectation-maximization. In: 2005 IEEE computer society conference on computer vision and pattern Recognition, Vol. 1. IEEE; 2005, p. 747–54.
[8] Přibyl B, Chalmers A, Zemcík P, Hooberman L, Cadik M. Evaluation of feature point detection in high dynamic range imagery. J Vis Commun Image Represent 2016;38:141–60.
[9] Barakat N, Hone AN, Darcie TE. Minimal-bracketing sets for high-dynamic-range image capture. IEEE Trans Image Process 2008;(10):1864–75.
[10] Debevec PE, Malik J. Recovering high dynamic range radiance maps from photographs. In: Proceedings of the 24th annual conference on computer graphics and interactive techniques. ACM Press/Addison-Wesley Publishing Co.; 1997, p. 369–78.
[11] Robertson M, Borman S, Stevenson R. Dynamic range improvement through multiple exposures. In: Proceedings 1999 international conference on image processing (Cat. 99CH36348), Vol. 3. 1999, p. 159–63.
[12] Robertson MA, Borman S, Stevenson RL. Estimation-theoretic approach to dynamic range enhancement using multiple exposures. J Electron Imaging 2003;12(2):219–28.
[13] Granados M, Ajdin B, Wand M, Theobalt C, Seidel H-P, Lensch HPA. Optimal HDR reconstruction with linear digital cameras. In: 2010 IEEE computer society conference on computer vision and pattern recognition. 2010, p. 215–22.
[14] Mann S, Picard RW. On being 'undigital' with digital cameras: Extending dynamic range by combining differently exposed pictures. In: PROCEEDINGS of IS&T. 1995, p. 442–8.
[15] Reinhard E, Heidrich W, Debevec P, Pattanaik S, Ward G, Myszkowski K. High dynamic range imaging: Acquisition, display, and image-based lighting. Morgan Kaufmann; 2010.
[16] Reinhard E, Stark M, Shirley P, Ferwerda J. Photographic tone reproduction for digital images. In: Proceedings of the 29th annual conference on computer graphics and interactive techniques. 2002, p. 267–76.
[17] Dufaux F, Le Callet P, Mantiuk R, Mrak M. High dynamic range video: From acquisition, to display and applications. Academic Press; 2016.
[18] Banterle F, Artusi A, Debattista K, Chalmers A. Advanced high dynamic range imaging. AK Peters/CRC Press; 2017.
[19] Eilertsen G, Mantiuk RK, Unger J. A comparative review of tone-mapping algorithms for high dynamic range video. In: Computer graphics forum, Vol. 36, no. 2. Wiley Online Library; 2017, p. 565–92.
[20] Ou Y, Ambalathankandy P, Takamaeda S, Motomura M, Asai T, Ikebe M. Real-time tone mapping: A survey and cross-implementation hardware benchmark. IEEE Trans Circuits Syst Video Technol 2021.
[21] Mai Z, Mansour H, Mantiuk R, Nasiopoulos P, Ward R, Heidrich W. Optimizing a tone curve for backward-compatible high dynamic range image and video compression. IEEE Trans Image Process 2010;20(6):1558–71.
[22] Debevec P, Gibson S. A tone mapping algorithm for high contrast images. In: 13th Eurographics workshop on rendering. Pisa, Italy: Citeseer; 2002.
[23] Drago F, Myszkowski K, Annen T, Chiba N. Adaptive logarithmic mapping for displaying high contrast scenes. In: Computer graphics forum, Vol. 22, no. 3. Wiley Online Library; 2003, p. 419–26.
[24] Ferradans S, Bertalmio M, Provenzi E, Caselles V. An analysis of visual adaptation and contrast perception for tone mapping. IEEE Trans Pattern Anal Mach Intell 2011;33(10):2002–12.
[25] Lischinski D, Farbman Z, Uyttendaele M, Szeliski R. Interactive local adjustment of tonal values. ACM Trans Graph 2006;25(3):646–53.
[26] Mantiuk R, Daly S, Kerofsky L. Display adaptive tone mapping. In: ACM SIGGRAPH 2008 papers. 2008, p. 1–10.
[27] Smith K, Krawczyk G, Myszkowski K, Seidel H-P. Beyond tone mapping: Enhanced depiction of tone mapped HDR images. In: Computer graphics forum, Vol. 25, no. 3. Wiley Online Library; 2006, p. 427–38.
[28] Mantiuk R, Mantiuk R, Tomaszewska A, Heidrich W. Color correction for tone mapping. In: Computer graphics forum, Vol. 28, no. 2. Wiley Online Library; 2009, p. 193–202.
[29] Mantiuk R, Seidel HP. Modeling a generic tone-mapping operator. In: Computer graphics forum, Vol. 27, no. 2. Wiley Online Library; 2008, p. 699–708.
[30] Eilertsen G, Mantiuk RK, Unger J. Real-time noise-aware tone mapping. ACM Trans Graph 2015;34(6):1–15.
[31] Rana A, Valenzise G, Dufaux F. Learning-based tone mapping operator for efficient image matching. IEEE Trans Multimed 2018;21(1):256–68.
[32] Debattista K. Application-specific tone mapping via genetic programming. In: Computer graphics forum, Vol. 37, no. 1. Wiley Online Library; 2018, p. 439–50.
[33] Rana A, Singh P, Valenzise G, Dufaux F, Komodakis N, Smolic A. Deep tone mapping operator for high dynamic range images. IEEE Trans Image Process 2019;29:1285–98.
[34] Comino M, Andújar C, Bosch C, Chica A, Muñoz-Pandiella I. Intensity-guided exposure correction for indoor LiDAR scans. In: XXX Spanish computer graphics conference. European Association for Computer Graphics (Eurographics); 2021, p. 15–8.
[35] Comino M, Andújar C, Bosch C, Chica A, Muñoz-Pandiella I. Neural colorization of laser scans. In: XXX Spanish computer graphics conference. CEIG 2021: MáLaga, Spain, September 22-24, 2021, European Association for Computer Graphics (Eurographics); 2021, p. 9–14.
[36] Comino M, Andujar C, Chica A, Brunet P. Error-aware construction and rendering of multi-scan panoramas from massive point clouds. Comput Vis Image Underst 2017;157:43–54.
[37] Liu YJ, Yu M, Li BJ, He Y. Intrinsic manifold SLIC: A simple and efficient method for computing content-sensitive superpixels. IEEE Trans Pattern Anal Mach Intell 2017;40(3):653–66.