Faculty of Sciences and Technology
University of Coimbra

Color Reproduction and Augmented Reality for Video-Based Surgical Navigation

Pedro Miguel Cerveira Rodrigues

Ph.D. thesis in Electrical and Computer Engineering mentored by Prof. Dr. João Pedro de Almeida Barreto and presented to the Department of Electrical and Computer Engineering, Faculty of Sciences and Technology, University of Coimbra

July, 2020

1 2 9 0

UNIVERSIDADE Đ
COIMBRA

Contents

Acronyms

AR augmented reality. 52, 55, 59

BTF brightness transfer function. 25–27

CRF camera response function. 3, 5–7, 9, 11, 13–15, 18–23, 25, 26, 29

CT computed tomography. 51, 52

DSLR digital single lens reflex. 36, 39, 44, 46–49

LUT look-up table. 11, 37, 38, 40–43, 45

MIP minimally-invasive procedure. 4, 7, 27, 28, 30

MRI magnetic resonance imaging. 52

OR operating room. 4, 7, 29, 50, 61

SfS shape-from-shading. 3, 27, 29

TKA total knee arthroplasty. 51–53, 57, 61

List of Figures

List of Tables

List of Algorithms

1 Introduction

The research work presented in this document relates to color reproduction and augmented reality in video-based surgical navigation. In this chapter, we will discuss the motivation behind the work and disclose technical information about this document, such as the outline, the notation used, and the doctoral program it relates to.

1.1 Thesis Outline

The remainder of this document is organized as follows:

Chapter 2 camera calibration

Chapter 3 color reproduction of display monitors

Chapter 4 6-D object pose estimation for augmented reality

1.2 Notation

Matrices are represented by symbols in a sans-serif font, *e.g.*, A. Vectors and vector functions are represented by bold symbols, and scalars are denoted by plain font letters, *e.g.*, $\mathbf{x} = (x, y)^\mathsf{T}$ and $\mathbf{f}(\mathbf{x}) = (f_x(\mathbf{x}), f_y(\mathbf{x}))^\mathsf{T}$. $f_x(\mathbf{x})$ denotes a scalar function. Sets are denoted by upper-case calligraphic letters, *e.g.*, $\mathcal{L}$.
Indices are denoted as underscripts and descriptive information is provided as superscripts.

1.3 Ph.D. Courses and Publications

1.3.1 Ph.D. Courses

During the curricular part of the Ph.D., the following disciplines were completed:

- advanced topics of operational research (grade of 18/20);
- cognitive vision systems (grade of 17/20);
- computational techniques for estimation, detection and identification (grade of 19/20);
- video-vigilance and biometrics (grade of 17/20).

The PhD course also involved the elaboration of a thesis project ...
The thesis project involved an oral presentation and it was evaluated in a public session with a final grade of 19/20. ...

1.3.2 Publications

[22] Elsevier Neurocomputing
[45] CVPR2015
[44] Granted Patent by USPTO
[47] Elsevier CVIU
[46] Submitted to Elsevier Displays
[43] IET-HTL

2 Camera Characterization with Invariance to Light and Vignetting

Camera response characterization determines how a camera transforms incoming light to create pixel values on a digital image. It has been a research topic for a few decades and its relevance is increasing because of the proliferation of computer vision algorithms in this domain. Applications include augmented reality, 3D reconstruction by shape-from-shading (SfS) [22], and color normalization across camera vendors [52].

The most simple form of camera response characterization is the estimation of what is usually known as the camera response function (CRF). This function is a camera model that, for each image channel independently, maps the value of the incoming light to pixel values. Currently, most methods in the literature only address this issue. The CRF may be adequate in some cases, such as for RAW images, where the only mapping is the light sensors response. However, for most cameras, the CRF does not fully describe the transformation [52]. It aims to linearize the response of each color channel independently, but does not describe the color transformations performed in-camera nor the color space of the camera. To fully describe a camera one must estimate a more complex model. Each vendor implements in-camera its own closed-source algorithm. See figure 2.1 for two images of similar scenes taken from two cameras with different responses. Even if two vendors follow the same color standard (*e.g.*, sRGB) the resulting images can be quite different. Thus, an universal camera model can be hard to establish. Several models have been proposed in the literature. Recently, an insightful work by [52] aimed to establish a more accurate camera model. Although it requires a large amount of images with the same camera, it was able to establish a better model for the in-camera processing, *i.e.*, from RAW values to RGB pixel values. The proposed model entails a CRF, a gamut mapping function, and a color transformation matrix. It also proved that the in-camera processing is not scene dependent as had been suggested in [11]. Missing from

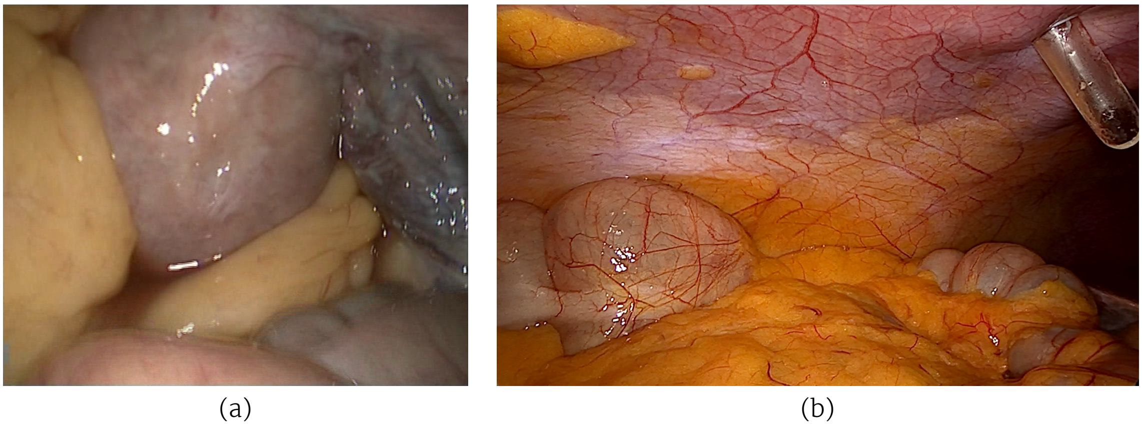


Figure 2.1 – Laparoscopic video frames of similar anatomical cavities from two minimally-invasive procedure (MIP) towers from different vendors. Note the difference in the color response of the two cameras on tissue with the similar color properties.

the Kim *et al*. model is the mapping from the actual incoming light on the sensor to the RAW camera-specific space. Describing the incoming light is usually done in a standard color space and the mapping between a camera-specific color space and a standard color space can be referred to as photometric or colorimetric calibration.

In contrast with the work of [52], we aim to achieve single-frame characterization, including photometric calibration, of digital cameras with generic vignetting and acquired under a generic illuminant. Currently, this is an unsolved problem in the literature. Papers addressing calibration in these conditions usually employ simple camera models with assumptions that do not hold in practice. Moreover, works using more accurate models, assume light is uniform on the scene and that vignetting either can be ignored or has a radial behavior.

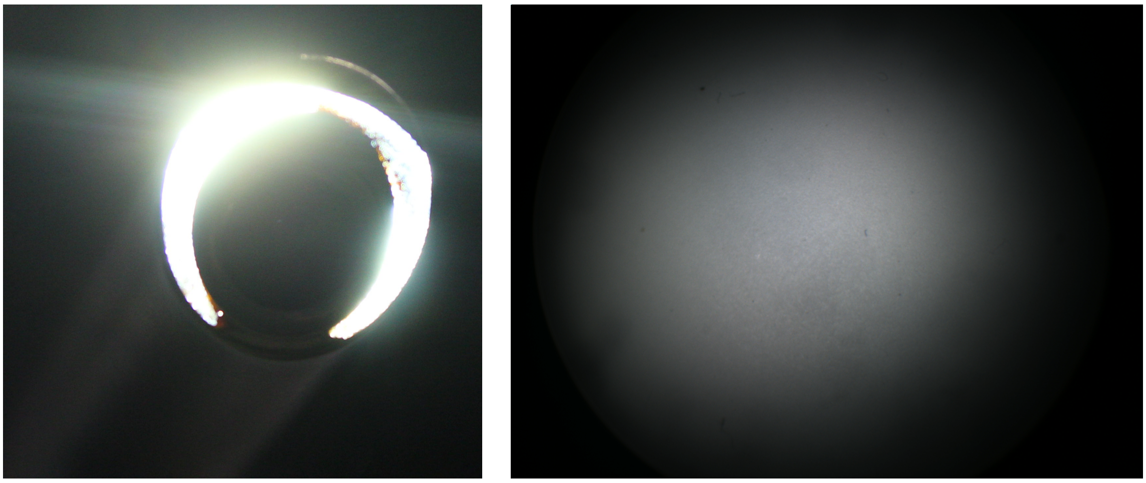
The set-up we address in this work is of particular interest in MIPs, where the previous assumptions do not hold. Moreover, cameras designed for MIPs are composed of inter-changeable components (camera, optics, and light). Thus, a medical professional must be able to make the calibration as the parts are mounted, on-the-fly, in the operating room (OR). Hence, the desire to develop a single-image method. Note that indoor calibration of generic cameras, such as cellphone cameras, also tend to violate the uniform illuminant assumption.

This work is an extension of what is described in [45]. As in that work, we propose a method that uses a surface with multiple regions of different colors as a calibration target, for which the albedo values are known in advance. Although the shape of the calibration surface is irrelevant for the estimation, all the experiments are carried using a planar checkerboard target similar to the one described in [2]. The reasons are twofold: the automatic segmentation and identification of the albedo regions is straightforward, and by combining our method with the geometric calibration approach proposed in [35] it is possible to fully model the camera from a single calibration frame. In addition to the work presented in [45], this document provides a deeper analysis and theoretical insight about the image formation physics, the camera models, and the necessary conditions for the method to work. It also extends the method that only was able to estimate the CRF, to also perform color response characterization. This extension brings two advantages: (a) it relaxes the assumption that the camera color response is in sRGB with positive impact in accuracy/repeatability of the CRF, and (b) it models the color response of the camera allowing for its calibration. Relaxing this assumption and characterizing the camera color response entails estimating a color transformation matrix in addition to the CRF. As a consequence of estimating a more accurate model, the theory behind the approach for dealing with generic lighting setups must be revisited. In addition, the number of albedos needed in the calibration target changes from two albedos to at least three albedos for the usual trichromatic cameras.

The current chapter will present a thorough experimental validation for different set-ups, a study on the influence of the number of images and colors involved in the calibration, and examples of practical applications using real endoscopic imagery. Our method will also be evaluated for non-endoscopic cameras and we will demonstrate its applicability to such cameras. The method of [62, 63] was identified as being the closest to our work and will be used for comparison purposes. Further information about their approach is given in section 2.1.2.

2.1 Related Work

Literature in classical CRF estimation is abundant and there are some well established methods. However, to the best of our knowledge, there is not a practical method that can cope with set-ups with near light from an illuminant with generic shape and/or generic vignetting (see figure 2.2). Moreover, most methods work with RAW images, without in-camera software being applied to the acquired pixel values, and thus can use simple camera models that will not generalize to RGB images. Most off-the-shelf cameras, however, cannot export RAW images, making this a very important limitation of these methods. The proposed approach relaxes this assumptions allowing for a more complete cam-



(a) (b)

Figure 2.2 – A photograph of the lens tip of an endoscopic camera (a) and a white planar scene imaged with an endoscopic camera (b).

Table 2.1 – State-of-the-art methods described in the introduction regarding camera response characterization. From left to right: the methods; weather the method can be used with a single-image; whether the method allows for flexible camera setting and/or pose throughout the calibration procedure; whether it can cope with near lighting setups; the camera model used (CRF – camera response function; M – color mapping; LUT – 3D look-up table); whether it performs photometric calibration. Note that photometric calibration requires the use of a calibration target and that single-image based methods do not require variable camera settings.

method	*single image*	*flexible pose/settings*	*near light*	*model*	*photometry*
photo collections [16, 30]	✗	✓	✗	CRF	✗
variable exposure [15, 36]	✗	✗	✓	CRF	✗
single-image [32, 38, 33, 61]	✓	✓	✗	CRF	✗
photometry [25, 20, 17]	✓	✓	✓	M	✓
Wu *et al.* [63]	✗	✗	✓	CRF	✓
Kim *et al.* [52]	✗	✗	✓	CRF+M+LUT	✗
Rodrigues and Barreto [45]	✓	✓	✓	CRF	✓
Rodrigues *et al.* [47]	✓	✓	✓	CRF+M	✓

era model.

In this section, we will provide an overview of the classical calibration (CRF estimation) and of the relevant works in the state of the art, and discuss why these methods cannot be applied in the scenarios tackled in this work. Table 2.1 provides a summary of the methods and their requirements.

2.1.1 Conventional Calibration

Typically, CRF estimation is performed using multiple registered images of the same static scene. For this, photo collections, *e.g.*, a panoramic shot, can be used to infer the CRF from the overlapping regions [16, 30]. In [16], authors assume a distant light model for the parametrization. As for [30], the work assumes that the vignetting is static and has a circular attenuation effect from the lens center. Moreover, the light is assumed to be static relative to the scene, whereas in our case the light moves with the camera.

Another possibility, and the one most often adopted, is using a perfectly static camera and a static scene to acquire images at different exposure values [15, 36], usually by varying the shutter speed. While this is a widely used approach, many off-the-shelf cameras do not allow to manually set the shutter speed and, in MIPs, it can be difficult to acquire multiple images in the same pose as these cameras are designed to be handheld and must be sterilized. Kim *et al.* [52] estimate a more complex camera model, but their method uses approximately 30 images, including same-pose images with different exposures and camera settings.

Some single-image approaches for CRF estimation for uniform light have also been proposed [32, 38, 33, 61]. These methods all have in common the assumption that the light is uniform on the regions being analyzed in each method, and this is hardly the case when dealing with near lighting (see figure 2.2).

As mentioned before, none of these methods work with more complete camera models. Instead they limit the camera model to the CRF, which either means that the authors are dealing with RAW data or that the cameras are not being fully characterized.

2.1.2 Near-Light Set-Ups

Only a small number of works explicitly deal with near light set-ups. In [63], the authors propose a method to estimate the CRF using 24 same-pose same-exposure images each of a single-color patch of known albedo. However, this approach cannot be done on-the-fly in the OR, which is a major disadvantage for application in MIPs. It is also not applicable in generic cameras for which the shutter speed cannot be manually set. Moreover, like the conventional calibration methods mentioned above, it has the drawback of using a camera model with just the CRF. The methods proposed in [25, 20, 17] can also cope with near-light set-ups. However, the authors assume the camera response is already in a linear space. Therefore, instead of performing CRF estimation, the authors use a 3 by 3 matrix to remap the RGB values of the camera.

2.1.3 Other Related Works

Works in the color stabilization field aim to match the color properties of two images with shared content [57, 19, 56, 58]. While this type of approach could potentially be used in our scenario, these works do not aim to characterize the camera. For example, in [58] the authors estimate what could be perceived as a CRF and in [56] they estimate a 3 by 3 matrix. However, the estimated models do not describe the camera in a canonical color space. Instead, these works aim to match the tonal properties of two or more images, so that the images could be perceived to be taken under similar circumstances, *i.e.*, similar lighting and camera settings.

2.2 Image Formation: Theory and Assumptions

In this section we will discuss the physics of image formation and the theory behind its modeling, as well as the assumptions that are made in this work. For a summary of the assumptions related to the image formation please see section 2.2.1.

The image formation model can be conceptually divided in two parts: the physics of the incoming light and the camera model. First, let us look at the physics part of the image formation. The sensor irradiance adopted in this work is

$$e(\mathbf{x}, \lambda) = v(\mathbf{x}) q(\mathbf{x}, \lambda) \rho(\mathbf{x}, \lambda) \tag{2.1}$$

as in [18, 21], where **x** are the pixel coordinates that correspond to the projection of a scene point, λ is the wavelength, v is the vignetting (a combination of natural, optical, mechanical, and pixel vignetting), q is the light component reflected from the scene, and ρ is the albedo (color). For a trichromatic camera, the generic image acquisition process is

$$\mathbf{d}\left(\mathbf{x}\right)=\mathbf{f}\left(\alpha\begin{bmatrix}\int e\left(\mathbf{x},\lambda\right)s_{\mathrm{R}}\left(\lambda\right)\,d\lambda\\\int e\left(\mathbf{x},\lambda\right)s_{\mathrm{G}}\left(\lambda\right)\,d\lambda\\\int e\left(\mathbf{x},\lambda\right)s_{\mathrm{B}}\left(\lambda\right)\,d\lambda\end{bmatrix}\right)\tag{2.2}$$

as in [64, 18], where $\mathbf{f}:\mathbb{R}^3\to\mathbb{R}^3$ is a generic camera model that comprises any non-linearities on the sensors and camera intrinsic processing, $\mathbf{d}=\left(d_{\mathrm{R}},d_{\mathrm{G}},d_{\mathrm{B}}\right)^{\mathsf{T}}$ is the acquired image, α is the exposure (a combination of sensor gains, shutter time, and aperture), $s_c\left(\lambda\right)$ is the spectral sensitivity of the sensors on channel $c\in\{\mathrm{R},\mathrm{G},\mathrm{B}\}$. The equations present in this work generalize for other cameras such as grayscale cameras and hyperspectral/multispectral cameras, but for an easier read we will use RGB cameras in this description.

Since we are interested in calibrating the camera and, therefore, transform image $\mathbf{d}$ to a new image in a canonical space, we are actually interested in modeling the inverse of the camera model $\mathbf{f}^{-1}$. Note that $\mathbf{f}$ is not guaranteed to have an inverse. In fact, it does not usually have one. However, $\mathbf{f}^{-1}$ is denoted here as the best possible approximation of the inverse of $\mathbf{f}$.

Furthermore, if one can assume that light component has the same spectrum across all $\mathbf{x}$, then

$$q\left(\mathbf{x},\lambda\right)=q_{\mathbf{x}}\left(\mathbf{x}\right)q_{\lambda}\left(\lambda\right).\tag{2.3}$$

In practice, this is the case for a single illuminant with constant spectrum, or for multiple illuminants all with the same spectra. This can also be assumed in a near light scenario where the intensity of the ambient light is negligible when compared to the near lights. Another requirement for this assumption to be case is that the bidirectional reflectance distribution function is the same for all $\mathbf{x}$. In that case, one can write

$$\mathbf{f}^{-1}\left(\mathbf{d}\left(\mathbf{x}\right)\right)=\alpha v\left(\mathbf{x}\right)q_{\mathbf{x}}\left(\mathbf{x}\right)\begin{bmatrix}\int q_{\lambda}\left(\lambda\right)\rho\left(\mathbf{x},\lambda\right)s_{\mathrm{R}}\left(\lambda\right)\,d\lambda\\\int q_{\lambda}\left(\lambda\right)\rho\left(\mathbf{x},\lambda\right)s_{\mathrm{G}}\left(\lambda\right)\,d\lambda\\\int q_{\lambda}\left(\lambda\right)\rho\left(\mathbf{x},\lambda\right)s_{\mathrm{B}}\left(\lambda\right)\,d\lambda\end{bmatrix}$$

or

$$\mathbf{f}^{-1}\left(\mathbf{d}\left(\mathbf{x}\right)\right)=\alpha v\left(\mathbf{x}\right)q_{\mathbf{x}}\left(\mathbf{x}\right)\boldsymbol{\rho}^{\mathrm{RAW}}\left(\mathbf{x}\right)\tag{2.4}$$

where $\boldsymbol{\rho}^{\mathrm{RAW}}=\left(\rho_{\mathrm{R}}^{\mathrm{RAW}},\rho_{\mathrm{G}}^{\mathrm{RAW}},\rho_{\mathrm{B}}^{\mathrm{RAW}}\right)^{\mathsf{T}}$ is the albedo of the scene as imaged under the light of spectrum q_{λ} in the RAW color space, *i.e.*, the color space of the camera sensors or, in other words, the color space before any in-camera processing.

As for the camera model, the adopted $\mathbf{f}^{-1}$ entails a strictly increasing function for each channel $g_c : \mathbb{R} \rightarrow \mathbb{R}$, the inverse CRF, followed by an invertible square matrix M^{m}, the color transformation matrix. Formally,

$$\mathbf{f}^{-1}(\mathbf{d}) = \mathsf{M}^{\mathrm{m}}\mathbf{g}(\mathbf{d}) \tag{2.5}$$

where $\mathbf{g}(\mathbf{d}) = (g_{\mathrm{R}}(d_{\mathrm{R}}), g_{\mathrm{G}}(d_{\mathrm{G}}), g_{\mathrm{B}}(d_{\mathrm{B}}))^{\mathsf{T}}$. While there has been some discussion on the validity of this model [24, 11], it has been used with good results and has been proven to be valid for most of the range of possible image values [52].

In this work, we are interested in performing photometric calibration which entails the mapping of the color space of the camera with a 3 by 3 matrix M^{s} that converts the values in a standard space related to the standard human observer, such as the sRGB color space, to the camera space. This matrix cannot fully describe this transformation (Luther-Ives condition [59]). However, it is a widely made approximation that is enough for practical purposes. Such matrix could also have other dimensions and, in fact may not be a square matrix, *e.g.*, the case where one wants to map a trichromatic camera to a single-channel space or map a hyperspectral camera to a trichromatic space. However, throughout this work only square matrices are considered. Let us rewrite equation (2.4) as

$$\mathbf{f}^{-1}(\mathbf{d}(\mathbf{x})) = u(\mathbf{x})\,\boldsymbol{\rho}^{\mathrm{RAW}}(\mathbf{x}) \tag{2.6}$$

where $u(\mathbf{x})$, what we will refer to as the albedo-normalized irradiance, is composed of exposure, vignetting, and the light effect on the scene, *i.e.*,

$$u(\mathbf{x}) = \alpha v(\mathbf{x})\,q_{\mathbf{x}}(\mathbf{x})\,. \tag{2.7}$$

Using the camera model from equation (2.5), the color standardization matrix as

$$\boldsymbol{\rho}^{\mathrm{sRGB}} = \mathsf{M}^{\mathrm{s}}\boldsymbol{\rho}^{\mathrm{RAW}}\,, \tag{2.8}$$

and defining the color transformation matrix as $\mathsf{M} = \mathsf{M}^{\mathrm{s}}\mathsf{M}^{\mathrm{m}} = (\mathbf{m}_{\mathrm{R}}, \mathbf{m}_{\mathrm{G}}, \mathbf{m}_{\mathrm{B}})^{\mathsf{T}}$ one can finally write the image formation model as

$$\mathsf{M}\mathbf{g}(\mathbf{d}(\mathbf{x})) = u(\mathbf{x})\,\boldsymbol{\rho}^{\mathrm{sRGB}}(\mathbf{x}) \tag{2.9}$$

in a standard color space, in this case sRGB.

Figure 2.3 provides a schematic overview of the image formation and of the calibration processes. In this figure, we show the light of an illuminant being

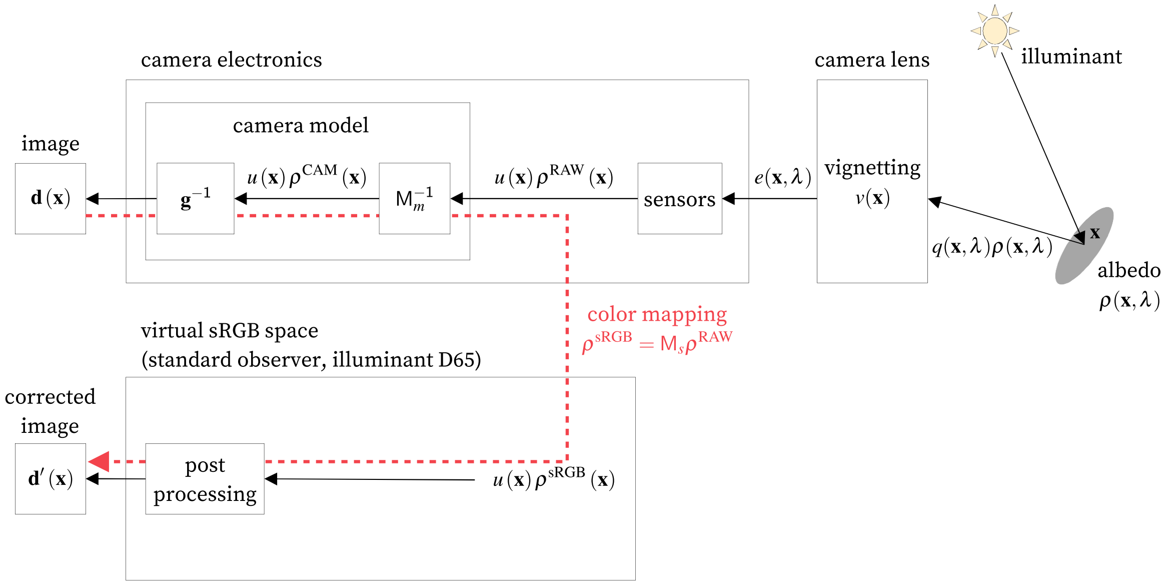

camera electronics
camera lens
illuminant
camera model
image
$\mathbf{d}(\mathbf{x})$
$\mathbf{g}^{-1}$
$u(\mathbf{x})\rho^{\mathrm{CAM}}(\mathbf{x})$
M_m^{-1}
$u(\mathbf{x})\rho^{\mathrm{RAW}}(\mathbf{x})$
sensors
$e(\mathbf{x},\lambda)$
vignetting
$v(\mathbf{x})$
$q(\mathbf{x},\lambda)\rho(\mathbf{x},\lambda)$
$\mathbf{x}$
albedo
$\rho(\mathbf{x},\lambda)$
color mapping
$\rho^{\mathrm{sRGB}} = \mathsf{M}_s\rho^{\mathrm{RAW}}$
virtual sRGB space
(standard observer, illuminant D65)
corrected image
$\mathbf{d}'(\mathbf{x})$
post processing
$u(\mathbf{x})\rho^{\mathrm{sRGB}}(\mathbf{x})$

Figure 2.3 – Schematic representation of the image formation process along with how the calibration procedure goes from the acquired image to the sRGB color space in order to create a corrected image. The calibration is represented by the dashed red arrow.

reflected from the scene and entering the camera to form an image. We show a schematic representation of how the equations change at each step and how $\boldsymbol{\rho}^{\mathrm{RAW}}$, $\boldsymbol{\rho}^{\mathrm{CAM}}$, $\boldsymbol{\rho}^{\mathrm{sRGB}}$ relate to each other. We also show how the calibration is used to create a new corrected image $\mathbf{d}'$ (dashed arrow).

Figure 2.4 provides a brief qualitative analysis to show how well the camera model that we will be using (*i.e.*, $\mathsf{M}\mathbf{g}(.)$) describes an actual camera. In this figure, we depict a comparison between two images of planar patches of a single albedo, where the images were taken in the same camera pose. Since the albedo is constant on each scene, the variations throughout the image $\mathbf{d}$ are due to the vignetting and the amount of reflected light from the scene, *i.e.*, the albedo-normalized irradiance $u(\mathbf{x})$. Also, since the camera pose and the illuminants are the same for the two images being compared, the scalar function $u(\mathbf{x})$ should be the same for the two images. Therefore, as long as the camera model is valid, isocurves on the two images being compared should be the coincident. They would have different values, but the same position. This is expected because the adopted camera model $\mathbf{f}^{-1}$ does not allow for constant values of $u(\mathbf{x})$ to be mapped to non constant values on the image side $\mathbf{d}(\mathbf{x})$. To test this hypothesis we represent isocurves of different color patches in the same plot for comparison. If the isocurves are coincident for all pairs of image channels and albedos, the adopted $\mathbf{f}^{-1}$ should be valid. The isocurves represented in figure 2.4 were defined by searching for pixels of same value on a specific image channel. The left column compares two low-saturation color-patch images to show that they are coincident, even across channels. On the right column we compare a low- with a high-saturation patch image. This shows that the isocurves of the two images are similar but not coincident. This is because, in this case, the camera model starts to be less accurate as the camera software tends to be more complex on the saturated values. Color saturation did not seam to be the only factor, as suggested in [52], since we were able to find two low-saturation patches with deviations in the isocurves and we could also find a pair of a low- and a high-saturation patches with coincident isocurves.

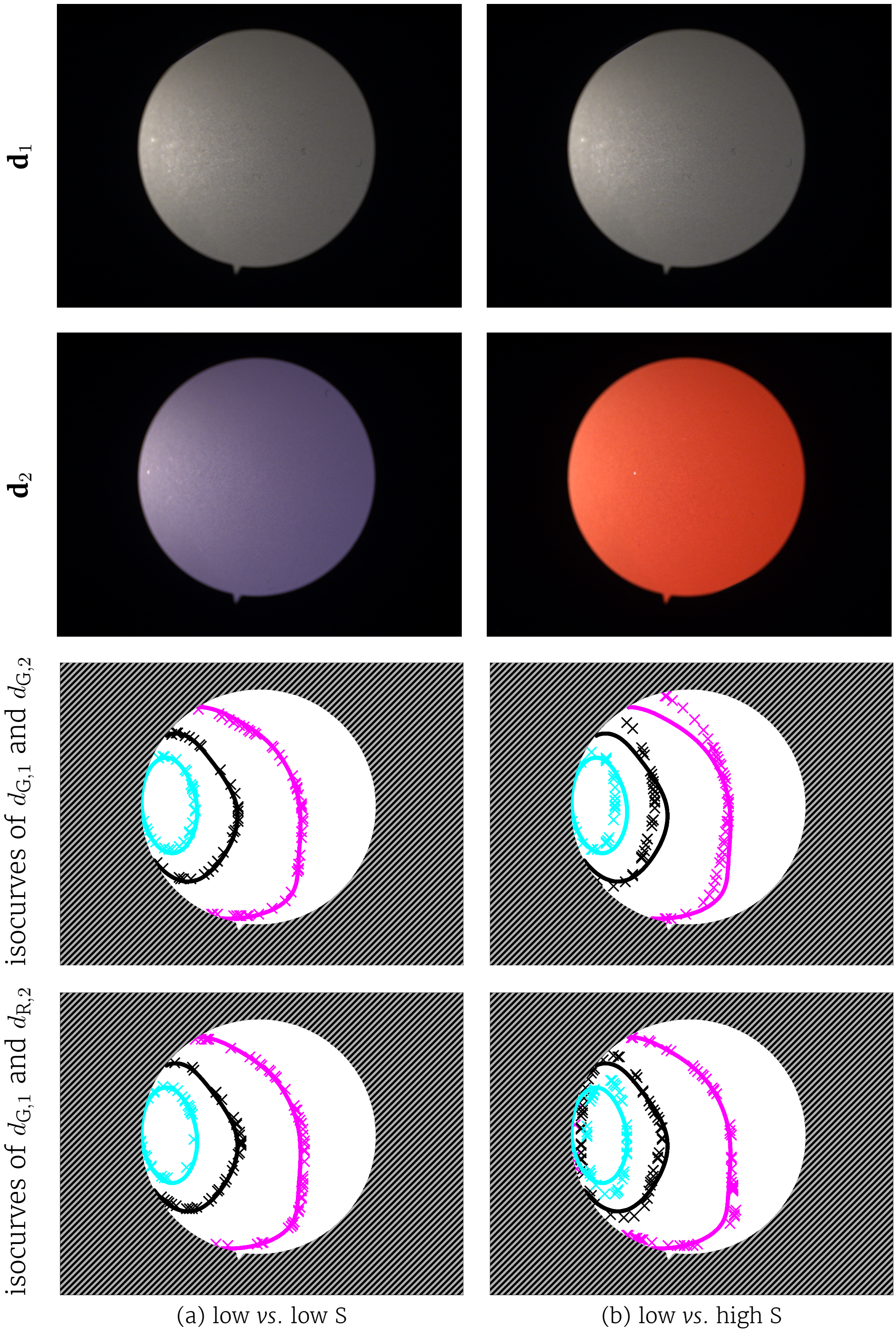
According to [52] a more accurate model for the camera must also include a gamut mapping function $\mathbb{R}^3 \to \mathbb{R}^3$, in the form of a 3D look-up table (LUT). However, we will not explore this model, since our goal is to perform the camera characterization with a single frame and to estimate a full 3D LUT we would need a wide range of data points. For comparison, to model a camera Kim *et al.* [52] use 30 images, including same-pose images with different exposures and camera settings.

2.2.1 Assumptions

In summary, the light component q is assumed to have the same spectrum for all $\mathbf{x}$, and the ambient light is assumed to negligible. This can be experimentally guaranteed by inserting the calibration target within a black box and performing the calibration there.

Regarding the assumptions about the calibration target, the surfaces do not have to be Lambertian. Yet, the BRDF must not change across albedos, which could happen if the surface properties change. In other words, we assume that the material has the same properties throughout the calibration target.

The exact number of colors that are needed for the calibration target varies with the model that is used. If only the CRF needs to be estimated or, in other words, the color transformation matrix M can be assumed to be diagonal, only two colors are needed. Three colors must be used if a full $\mathsf{M}_{3\times 3}$ is to be estimated. For a matrix of generic size the number of colors needed is equal to the number of columns of the matrix.



(a) low *vs*. low S

(b) low *vs*. high S

Figure 2.4 – Comparison of the isocurves of real images acquired using an endoscopic lens. The columns show the comparison between images of: (a) two low-saturation color patches, and; (b) a low-saturation and a high-saturation color patches. From top to bottom: the color-patch images being compared, the comparison between isocurves of the green channels of the two patches (lines — first image; crosses — second image), and the comparison between the isocurves of the green channel of the first patch with the red channel of the second patch (lines — first image; crosses — second image). See text for additional insight.

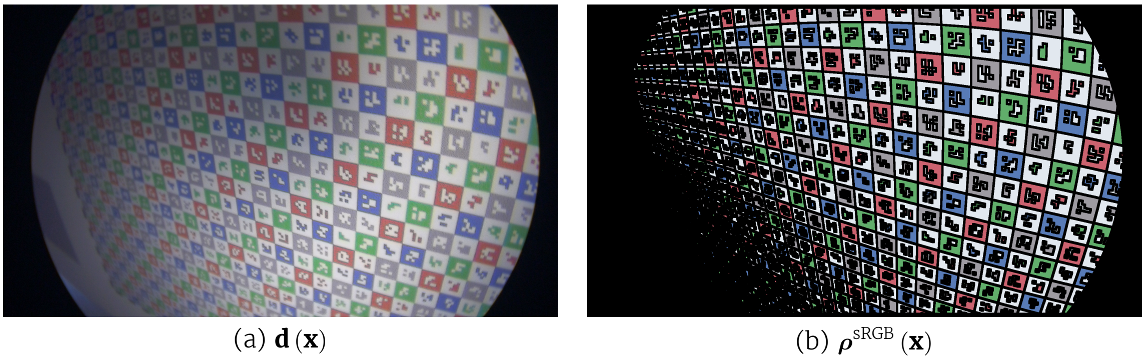

(a) $\mathbf{d}(\mathbf{x})$ (b) $\boldsymbol{\rho}^{\text{sRGB}}(\mathbf{x})$

Figure 2.5 – An image of a 5-color calibration grid (a) and the corresponding segmentation into regions of constant albedos (b) color-coded with the true sRGB values. Black pixels occur when albedo information is not available.

2.3 Camera Characterization

In this section we will discuss a preliminary work [45] and how that work can be extended to estimate both the CRF $\mathbf{g}$ and the color transformation matrix M. For an easier reformulation for other types of cameras (*e.g.*, different number of channels) and other color spaces, the channels of the acquired image will be indexed by the letter c and the elements of $\boldsymbol{\rho}^{\text{sRGB}}$ (the channels of the corrected image) will be indexed by the letter s. But note that usually $s \equiv c$, because, for most cases, one wants to map from an RGB space to sRGB, which have equivalent number of channels and channel names.

In [45], we propose the use of a single-image of a two-albedo (white and gray) target with known albedos to estimate the CRF. In both the preliminary work and what we propose with this work, we must perform estimation of isocurves, which will be explained in section 2.3.2. While the preliminary work only uses two sets albedo patches, the method proposed in this work was generalized to include more albedos so that the isocurves of u can be determined more accurately.

2.3.1 Segmentation

The proposed approach lies on the assumption that we have a scene of a multiple-color surface. We have used a planar multiple-color CALTag grid [2] geometrically calibrated with the method proposed in [35]. We do not need it to be planar nor a grid for the framework to succeed, but we do need to segment the scene into regions of constant albedo. We have chosen this grid to be able to perform both the geometric and photometric calibrations with a single image. The fact that this type of grid has tags in each square also allows for easier localization of each specific square and to ensure that a specific tag corresponds to a specific albedo.

Having used the method proposed in [35] to obtain the geometric calibration of the scene and the camera, the segmentation is straightforward. Since the positions on the scene plane of every fiducial marker and grid corner are known or can be easily estimated, we can warp the grid to the image plane. This warped image is itself our segmentation.

A morphological erosion is then performed to the segmented regions of each albedo to avoid problematic albedo boundary regions. On these regions, it is a possibility that the image values are influenced by optical and motion blur, chromatic aberrations, and demosaicing inaccuracies. Segmentation inaccuracies could also be another problematic factor on boundary regions.

Figure 2.5 shows a CALTag grid and its segmentation.

2.3.2 Isocurves Estimation

To use a single image for camera response characterization, we seek to find, within the same image, pairs of pixels where one could write equations invariant to the vignetting and the light effect. On a single-image approach, one cannot expect to find regions where both are constant without modeling them. However, in fact, we do not need to be invariant to both effects, only to their joint effect $u(\mathbf{x})$. In this way, we are able to build a system of equations where the only unknowns are the CRF and the albedos, without making assumptions about the vignetting or the light behavior on the scene. This is of crucial importance for our set-up, since the vignetting is not always central (as with most set-ups) and the lights are at close range, are typically not punctual (see figure 2.2), and may not be isotropic [14].

Let us define the albedos in yet another color space, the color space of the image $\mathbf{d}$, as $\boldsymbol{\rho}^{\mathrm{CAM}}$ such that

$$\boldsymbol{\rho}^{\mathrm{RAW}} = \mathsf{M}^{\mathrm{m}} \boldsymbol{\rho}^{\mathrm{CAM}} . \tag{2.10}$$

This color space differs from the RAW color space due to in-camera processing (see figure 2.3).

The particular case where the color transformation matrix M^{m} is approximated by a diagonal matrix has been addressed in [45], and we will build on this approach to eliminate this assumption.

For an easier understanding of the equations that follow, it is helpful to think of $u(\mathbf{x})$ as being the image that reaches the camera sensors but without any color information (whether from light sources or from the actual scene). By definition, an isocurve of $u\left(\mathbf{x}\right)$ will contain points $\mathbf{x}$ with the same value of $u\left(\mathbf{x}\right)$. Note that the points in an isocurve of $u\left(\mathbf{x}\right)$ may correspond to pixels with different values in the actual image $d\left(\mathbf{x}\right)$ since albedo information is present there. With this in mind and using equations (2.8), (2.9) and (2.10), if M is invertible, one can write

$$g_c\left(d_c\left(\mathbf{x}\right)\right) = u\left(\mathbf{x}\right)\rho_c^{\text{CAM}}\left(\mathbf{x}\right) \tag{2.11}$$

and, on the ith isocurve of u,

$$u(\mathbf{x}_j) = \kappa_i = \frac{g_c(d_c(\mathbf{x}_j))}{\rho_c^{\text{CAM}}(\mathbf{x}_j)}, \quad j \in \mathcal{L}_i \tag{2.12}$$

where $\mathcal{L}_i$ is the set of pixels crossed by isocurve i and κ_i is a constant (the value of the isocurve). Such is true for all c and for whatever albedo is crossed by the isocurve. If a curve i passes through multiple albedos one will have, for an albedo pair, $\boldsymbol{\rho}_n$ and $\boldsymbol{\rho}_{n'}$,

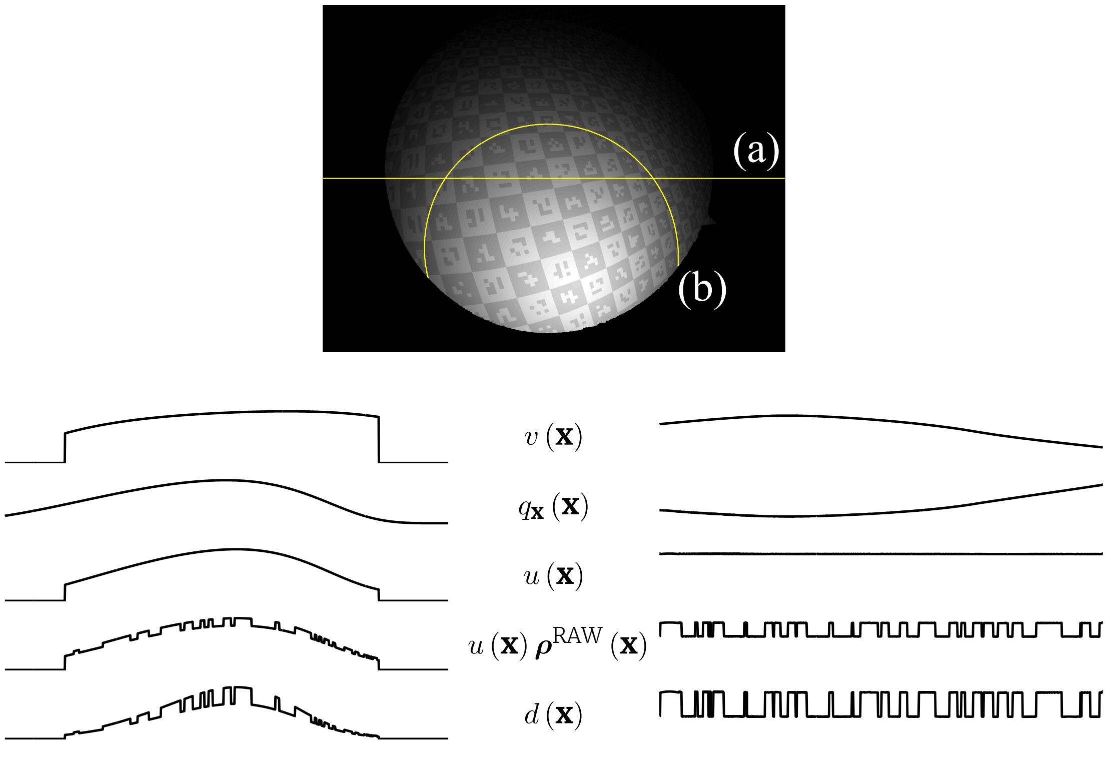
$$\rho_{c',n'}^{\text{CAM}} g_c(d_c(\mathbf{x}_j)) = \rho_{c,n}^{\text{CAM}} g_{c'}(d_{c'}(\mathbf{x}_k)), \quad j \in \mathcal{L}_i \cap \mathcal{A}_{\boldsymbol{\rho}_n}, k \in \mathcal{L}_i \cap \mathcal{A}_{\boldsymbol{\rho}_{n'}} \tag{2.13}$$

where $\mathcal{A}_{\boldsymbol{\rho}_n}$ is the set of points with a specific albedo $\boldsymbol{\rho}(\mathbf{x}) = \boldsymbol{\rho}_n$. Equation (2.13) will then be used for the single-image CRF estimation. Note that $c, c' \in \{\text{R}, \text{G}, \text{B}\}$ and thus c and c' are both channel indices. c' is used to show that the equations can be written by combining different channels or different albedos.

See figure 2.6 for more intuition on the image formation process and how it relates to the isocurves of $u\left(\mathbf{x}\right)$. This figure shows a synthetic grayscale image and each individual component that form the image. Figure 2.6(a) shows how the vignetting $v(\mathbf{x})$ and the light component $q(\mathbf{x})$ combine to form the albedo-normalized irradiance $u(\mathbf{x})$. It then shows what $u(\mathbf{x})$ combined with the albedo looks like and, finally, what a CRF does to that combination to create the final image. Figure 2.6(b) shows the same process for an isocurve of $u(\mathbf{x})$.

Since $u(\mathbf{x})$ is not known, we need to evaluate how its isocurves behave on the image $\mathbf{d}(\mathbf{x})$. From (2.12), it is clear that, for a given albedo, an isocurve in the sensor irradiance is also an isocurve in the image $d(\mathbf{x})$. In addition, along an isocurve of $u(\mathbf{x})$, $\mathcal{L}_i$, the image values form a piecewise constant function (with a different constant value for each albedo).

In the image space we have a set of isocurves for each albedo. However, the isocurves of $\mathbf{d}(\mathbf{x})$ for each albedo are the same and equal to the isocurves of $u(\mathbf{x})$, except for its value. The same is true for the one albedo and different image



(a) (b)

Figure 2.6 – Values (arbitrary units) of each image component showing the formation process of an image of a checkerboard along (a) a horizontal line and (b) an isocurve of the albedo-normalized irradiance $u\left(\mathbf{x}\right)$. For this example we used a synthetic grayscale image of a two-albedo checkerboard. Please refer to the text, specifically equations (2.4) and (2.6), for details on each image component.

color channels. Thus, to find the isocurves of $u(\mathbf{x})$, it is reasonable to fit a single surface $\mu_{c,n}$ to the image along one albedo $\boldsymbol{\rho}_n$ and channel c. If there is a single color suitable to fit μ across the whole calibration target, then a single color may be used. However, for better results, all albedos and all channels can be used.

Let us approximate the image along one of the albedos n and channel c by a generic model $\mu_{c,n}$ where the isocurves are known or can easily be extracted. We can write for two albedos on the image space

$$d_c(\mathbf{x}_j) \sim \mu_{c,n}(\mathbf{x}_j), \quad j \in \mathcal{A}_{\boldsymbol{\rho}_n} \tag{2.14a}$$

$$d_{c'}(\mathbf{x}_k) \sim \mu_{c',n'}(\mathbf{x}_k), \quad k \in \mathcal{A}_{\boldsymbol{\rho}_{n'}} . \tag{2.14b}$$

From before, we know that the isocurves of $\mu_{c,n}(\mathbf{x})$ and $\mu_{c',n'}(\mathbf{x})$ will have the same shape as the ones in $u(\mathbf{x})$ but with different values. The shape of the surfaces represented by the models are different, since the step between isocurves varies from one formulation to the other, but the isocurves are the same. One can show that the two models are related by

$$d_{c'}(\mathbf{x}_k) \sim \mu_{c',n'}(\mathbf{x}_k) \tag{2.15a}$$

$$g_c^{-1}\left(\frac{\rho_{c,n}^{\text{CAM}}}{\rho_{c',n'}^{\text{CAM}}} g_{c'}(d_{c'}(\mathbf{x}_k))\right) \sim \mu_{c,n}(\mathbf{x}_k) \tag{2.15b}$$

$$h_{c,c',n,n'}(d_{c'}(\mathbf{x}_k)) \sim \mu_{c,n}(\mathbf{x}_k), \quad k \in \mathcal{A}_{\boldsymbol{\rho}_{n'}} \tag{2.15c}$$

where $h_{c,c',n,n'}$ is a positive and monotonically increasing function that is used to transform the model $\mu_{c',n'}$ into the model $\mu_{c,n}$. This function h is the equivalent of having a gain for each isocurve value for the points of the albedo $\boldsymbol{\rho}_{n'}$ on channel c', to be able to use only the model $\mu_{c,n}$ for both albedos using equations (2.14a) and (2.15c).

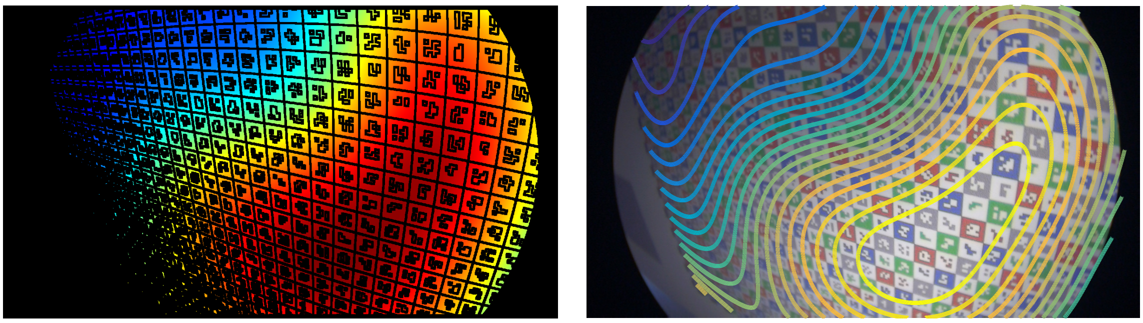
We have used a polynomial model as $\mu_{c,n}$, *i.e.*,

$$\mu_{c,n}(\mathbf{x};\mathbf{p}) = \hat{\mathbf{x}}^{\mathsf{T}}\mathbf{p},$$

where $\mathbf{p}$ are the polynomial coefficients and

$$\hat{\mathbf{x}} = \left(1, x_1, x_2, x_1^2, x_1 x_2, x_2^2, \dots\right)^{\mathsf{T}}.$$

The isocurves can then be extracted as the level sets of the polynomial. This is done by searching the polynomial function for points $\mathbf{x}$ where the polynomial is constant.



(a)

(b)

Figure 2.7 – A depiction of (a) the estimated $\mu_{c,n}(\mathbf{x})$ (a surface with the same isocurves as the albedo-normalized irradiance $u(\mathbf{x})$) and (b) the calibration grid image superposed with some of the isocurves.

As for the linear system of equations to be solved, for predefined c and n (*e.g.*, green channel and white albedo), it is defined as

$$\forall j \in \mathcal{A}_{\boldsymbol{\rho}_n}, \forall c', \forall n', \forall k \in \mathcal{A}_{\boldsymbol{\rho}_{n'}} : \begin{bmatrix} \hat{\mathbf{x}}_j^\top & \mathbf{0}^\top \\ \hat{\mathbf{x}}_k^\top & -\mathbf{s}^\top(d_{c'}(\mathbf{x}_k)) \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ \mathbf{h}_{c,c',n,n'} \end{bmatrix} = \begin{bmatrix} d_c(\mathbf{x}_j) \\ 0 \end{bmatrix} \tag{2.16}$$

where $\mathbf{h}$ is a discrete array version of the function h and $\mathbf{s}(n)$ is sparse vector with a single value of 1 on the element n. This is solved by minimization of the least squares with quadratic programming. We also constrain the various instances of $\mathbf{h}$ to be monotonically increasing by introducing additional equations that force the local derivatives to be positive. The polynomial used was of the 6th order (28 parameters). However, this can be altered to meet the requirements of other calibration scenes.

An example of the determined isocurves can be observed in figure 2.7.

2.3.3 Model Estimation

Having determined the isocurves, to estimate the camera model one would want to minimize the residuals derived from equation (2.13), and equations (2.8) and (2.10), as

$$\forall i, \forall s, \forall s', \forall n, \forall n' : e_{i,n,n',s,s'} = \rho^{\text{sRGB}}_{s',n'} \mathbf{m}_s^\top \mathbf{g}\left(\mathbf{d}_{i,\boldsymbol{\rho}_n}\right) - \rho^{\text{sRGB}}_{s,n} \mathbf{m}_{s'}^\top \mathbf{g}\left(\mathbf{d}_{i,\boldsymbol{\rho}_{n'}}\right) \tag{2.17}$$

where $\mathbf{d}_{i,\boldsymbol{\rho}}$ is the median value of the image values on isocurve i and albedo $\boldsymbol{\rho}$. However, this a nonlinear programming problem that can be hard to solve and time-consuming.

In [45], M is assumed to be diagonal and thus the problem can be estimated per channel using convex optimization by minimizing residuals of the form

$$\forall i, \forall s, \forall n, \forall n' : e_{i,n,n',s} = \rho^{\text{sRGB}}_{s,n'} g_c \left(d_{c,i,\boldsymbol{\rho}_n} \right) - \rho^{\text{sRGB}}_{s,n} g_c \left(d_{c,i,\boldsymbol{\rho}_{n'}} \right) \tag{2.18}$$

with $s = c$ as $\boldsymbol{\rho}^{\text{sRGB}} = \boldsymbol{\rho}^{\text{CAM}}$.

In this work, we propose the estimation of the CRF and M in an iterative process of two convex-optimization steps. The CRF is initially assumed to be linear, *i.e.*, $\forall c : g_c(d) = d$. Then, in each iteration, M is computed in a convex optimization framework using equations derived from equation (2.9) as

$$\forall \mathbf{x} : \left[\boldsymbol{\rho}^{\text{sRGB}} \left(\mathbf{x} \right) \right]_{\times} \mathsf{M} \mathbf{g} \left(\mathbf{d} \left(\mathbf{x} \right) \right) = \mathbf{0} \tag{2.19}$$

where $[\cdot]_{\times}$ denotes the skew-symmetric matrix. This is followed by the estimation of $\mathbf{g}$, which can be done with equations very similar way to equation (2.18). However, for robustness we have decided to introduce inter-channel equations. In other words, instead of only combining equations for different albedos (where $u(\mathbf{x})$ is constant) to exclude $u(\mathbf{x})$ from the equations, we combined equations from different channels as well.

The albedos are updated using equation (2.10) and the values used for M are the ones computed on the previous step. The residuals to minimize then become

$$\forall i, \forall c, \forall c', \forall n, \forall n' : e_{i,n,n',c,c'} = \rho^{\text{CAM}}_{c',n'} g_c \left(d_{c,i,\boldsymbol{\rho}_n} \right) - \rho^{\text{CAM}}_{c,n} g'_c \left(d_{c',i,\boldsymbol{\rho}_{n'}} \right) . \tag{2.20}$$

At the end of each iteration, a metric is computed to ensure the method is converging to a solution. The metric used was the average angle between the albedo vectors $\boldsymbol{\rho}^{\text{sRGB}}$ and their respective corrected image pixel $\mathsf{M}\mathbf{g}\left(\mathbf{d}\right)$, which should be co-linear. The angle θ, as computed between the vectors $\boldsymbol{\rho}^{\text{sRGB}}$ and $\mathsf{M}\mathbf{g}\left(\mathbf{d}\right)$, was used instead of an euclidean distance or a traditional color metric because the estimation is performed up-to-scale due to $u\left(x\right)$. This iterative process continues until there is no longer improvement on the metric or a fixed number of iterations is reached (50 iterations).

This approach is scalable with additional images. Even images with different exposures, different poses, and changes in the vignetting (due to changes of zoom and/or aperture) can be used to augment the number of equations. Additional images would provide additional equations that can be grouped together to potentially achieve a more robust estimation.

There is no need for both the light to be non-uniform on the scene and the vignetting to be strong. Either one is sufficient and, while the method benefits from these effects, neither is mandatory. If the calibration albedos cover the range of possible values, good results can be obtained as long as there is a slight variability enough for isocurves to be obtainable (for instance, a scene under sunlight with a faint shadow).

2.3.4 Parametrization

Since no assumption is made on the form of the CRF $\mathbf{g}$ with the isocurves approach, one is not bound to a specific CRF model. In fact, equation (2.20) can rewritten using specific parametrization for each g_c (*e.g.*, polynomial) or in a non-parametric way. In this work, we use a non-parametric formulation for the CRF $\mathbf{g}$.

The isocurves may not define equations for all values in the 0-255 range of image pixel values for the CRFs $g_c\left(d_c\right)$. Therefore, after estimating the camera model with non-parametric CRFs, these must be handled to fill values that might be missing, especially in the larger and lower values of this range. With this in mind, the CRF is parametrized with the Empiric Model of Response [23], a linear basis model obtained by using principal component analysis in real CRFs. Please check algorithm 2.1 for a summary of the steps of the camera characterization procedure.

Algorithm 2.1 Camera characterization algorithm.

```
image acquisition
geometric calibration
segmentation
estimation of $h(\mathbf{x})$ — equation (2.16)
level sets
2-step estimation of $\mathsf{M}$ and $\mathbf{g}$, the camera model — equations (2.19) and (2.20)
parametrization of $\mathbf{g}$
```

2.4 Experimental Validation

For the experimental validation we evaluated the color accuracy, CRF accuracy, and CRF repeatability using endoscopic and non-endoscopic set-ups.

Three endoscopic datasets were acquired along with a cellphone camera and a monochromatic camera to showcase the applicability to generic cameras and single-channel cameras:

1. PointGrey Flea3 (FLIR Systems, Wilsonville, OR) CMOS camera with a 30° Stryker (Stryker Corporation, Kalamazoo, MI) endocopic lens and a Smith & Nephew (Smith & Nephew plc, United Kingdom) light source connected through the built-in light guide

2. PontGrey Grasshopper2 CCD camera with a laparoscopic lens and an external lamp light

3. Sentech (Sentech co. ltd., Kanagawa Prefecture, Japan) CMOS camera with a 30° Dyonics (Smith & Nephew plc) endocopic lens and a light source connected through the built-in light guide

4. Cellphone (Samsung S6 Edge - Samsung Electronics co. ltd., South Korea) CMOS camera with the built-in LED flash turned on

5. PointGrey Dragonfly monochromatic CCD camera with a lamp as the light source

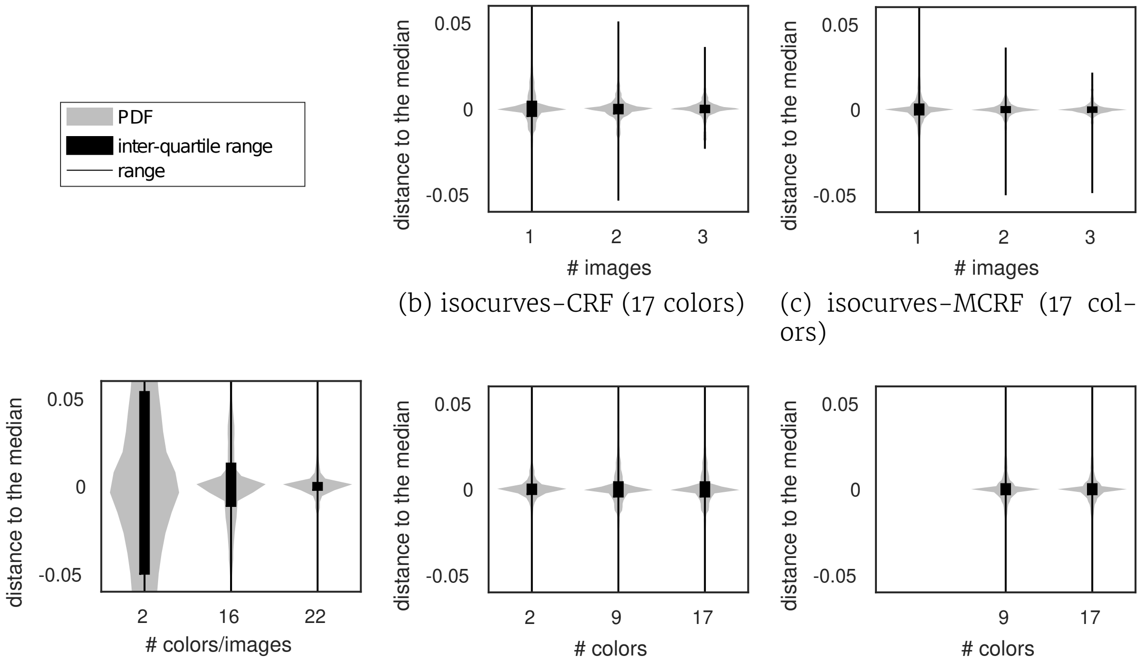
The CALTag checkerboard albedos were obtained with a iWave WR10 (iWave Systems Technologies, Bangalore, India) colorimeter.

Throughout the experimental results we compare the method proposed in this work (referred to as **isocurves-MCRF**) with the method presented in [45] (**isocurves-CRF**) and a fourth method (**direct-M**) where only M is estimated according to equation (2.19). The latter is essentially the first step of the iterative 2-step optimization proposed before. In it the CRF is assumed to be linear and thus no isocurves are necessary.

We also compare our methods to our implementation of the Wu *et al.* [63] approach, which requires the acquisition of same-pose same-exposure images of multiple single-albedo patches with known albedo values. Since the camera pose is static and the exposure is fixed, the only quantity that changes between images is the albedo. The authors assume that M is diagonal and they estimate each g_c function independently. To allow the comparison to this work, for each dataset that we acquired, in addition to the CALTag chart images, a sequence of images of the X-Rite (X-Rite, Inc., MI) ColorChart was acquired under the circumstances required by the method.

For all methods, the acquisitions were performed with ambient light present. There were no other illuminants close to the calibration scene, but there was indirect sunlight present in the scene.

The computational time of the isocurves-MCRF method was of approximately 12 seconds for 17-albedo 1600x1200 image using a MATLAB/C++ code developed



(b) isocurves-CRF (17 colors)

(c) isocurves-MCRF (17 colors)

(d) Wu *et al.*

(e) isocurves-CRF (1 image)

(f) isocurves-MCRF (1 image)

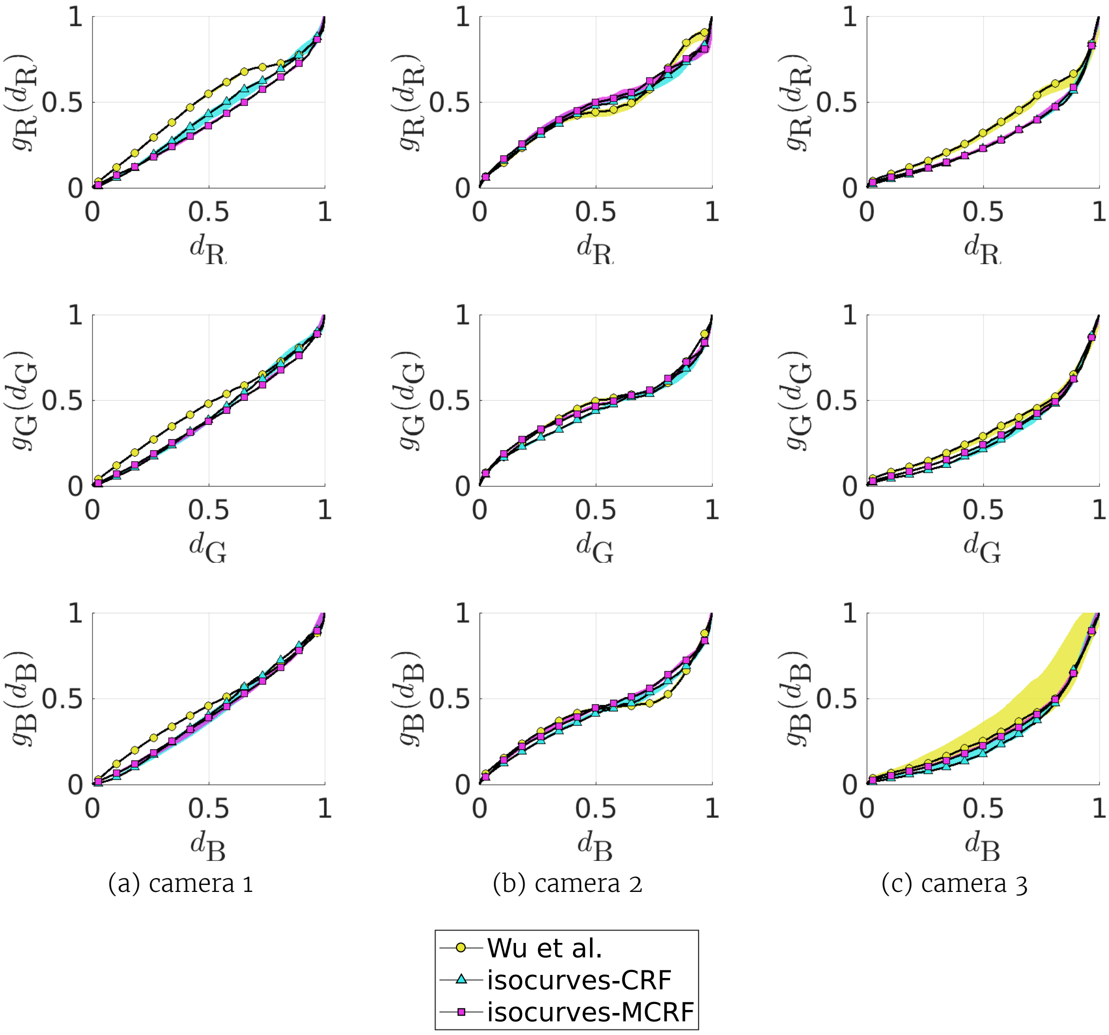
Figure 2.8 – Repeatability of the CRF estimation for the different approaches. The plots show the distribution and the boxplot of the distance to the median CRF. See text for more details on the metric.

for research purposes only and that has not been optimized for speed.

2.4.1 Color Accuracy and Repeatability

To assess the repeatability of the CRF estimation we present the histograms of the distance to the median CRF. For such metric, we performed the camera model estimation for all cameras and all poses. Since the camera models are estimated up to a global scale factor, the relative factors must be estimated for a fair comparison. Thus, for each camera, the CRFs of the red, green, and blue channels are aligned across the estimations on different camera poses, by estimating a single global scale factor for each pose. Then, the median CRF across poses is computed and subtracted from the estimated CRFs. The violin plot of figure 2.8, shows the distributions and boxplots of these differences from all datasets, all poses, all three channels, and all values in the pixel value range. Figure 2.9 shows the median and the inter-quartile range across different poses for the CRF estimation.

Using less calibration images reduces drastically the repeatability of the Wu *et al*. method, as shown by the distributions of figures 2.8 and 2.9. For our method, these results show that additional images and additional albedo regions only



(a) camera 1 (b) camera 2 (c) camera 3

Figure 2.9 – Median and inter-quartile range for the inverse CRFs for the (top to bottom) red, green, and blue channels of estimated for each dataset. Wu *et al.* approach was computed with 22 colors/images and the single-image approaches were computed with a single image of a 17-color target.

marginally improves repeatability. However, good repeatability is already obtained with a single image and fewer colors.

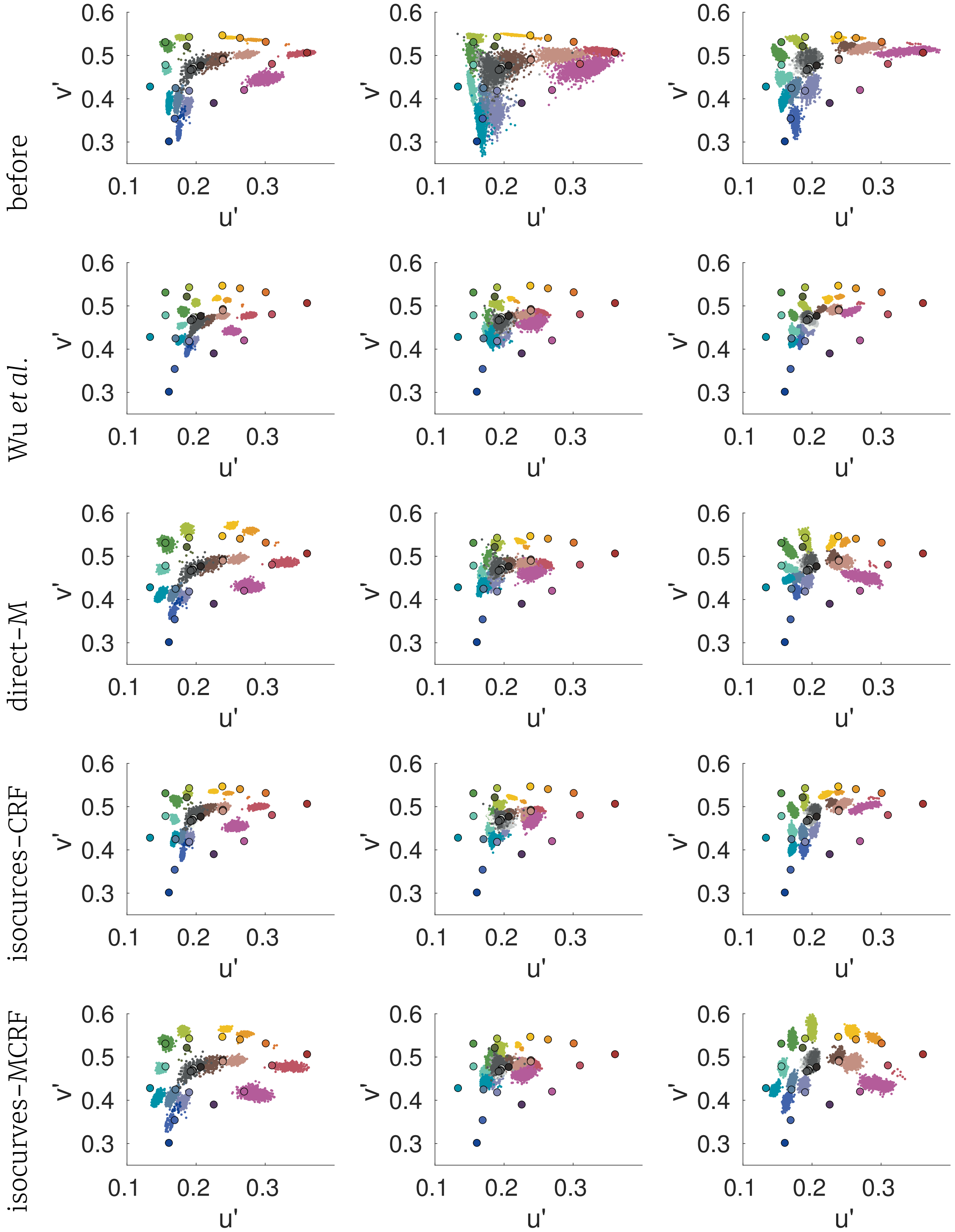
To evaluate color accuracy we use two metrics: θ, computed as the angle between the albedo vectors $\boldsymbol{\rho}^{\text{sRGB}}$ and their respective corrected image pixel $\mathsf{M}\mathbf{g}(\mathbf{d})$; and $\Delta_{u'v'}$, usually known as chromaticity distance. The latter is the euclidean distance in the $u'v'$-space, which can be obtained directly from the sRGB. Note that more standard color metrics like ΔE^{*}_{00} [53] cannot be used because the colors are being compared up-to-scale, due mainly to the near light on the scene. Since the light is not constant on the scene, as it would be in a direct sunlight set-up, the pixel values cannot be compared without estimating a scale factor for each pixel. Another reason is that we are estimating the camera model up-to-scale, therefore at least one global scale factor would need to be further estimated for color comparison. Both the θ and the $\Delta_{u'v'}$ provide comparison between colors with invariance to their ”lightness”.

Color accuracy was evaluated using the X-Rite ColorChart. The single-albedo images were used for testing the camera calibration performed with the isocurves-MCRF method, the isocurves-CRF method, and the Wu *et al*. method [63]. Wu *et al*. method was evaluated using a cross-validation approach. Pixels with values in the lower 10% of the 0-255 range have been discarded as these will be greatly affected by noise and thus will generate metrics that do not necessarily represent the data. Moreover, perceptually, the color mapping of the lower values is less important. For qualitative evaluation of the color accuracy, figure 2.10 presents the test pixel values after being corrected with the estimated model against the ground truth chromaticity from the X-Rite ColorChart. For sake of comparison, we also show the values of the original image without calibration, *i.e.*, simply assuming that camera is already in sRGB.

Table 2.2 shows a summary of both the CRF repeatability and the color metrics. As already shown in figure 2.8 and confirmed in table 2.2, in terms of CRF repeatability, the isocurves-MCRF method presented here is able to achieve similar results to the 22-image approach.

Although Wu *et al*. approach linearizes the color space by estimating a CRF, it performs worse, in terms of color accuracy, than the other methods evaluated here. In fact, like the isocurves-CRF method, it even shows to have worse color accuracy than the original image. This happens because, in both methods, M is assumed to be diagonal.

The estimation using the direct-M approach, *i.e.* the estimation of M assuming that $\forall c : g_c$ are linear, is purely color driven. This explains the fact that the



(a) camera 1 (b) camera 2 (c) camera 3

Figure 2.10 – Ground truth color points and respective corrected image pixel values in the chromaticity space. Note that Wu *et al*. was tested using 22 images and 22 colors, and the other methods were tested using one image and 17 colors.

Table 2.2 – Metrics of chromaticity distance and repeatability for the camera characterization methods across the three datasets. The table shows the average and standard deviation for both the $\Delta_{u'v'}$ and the θ metrics. The average inter-quartile range (IQR) metric used to assess repeatability is also presented.

	# albedos / # images	$\Delta_{u'v'}$	θ	IQR
before	– / –	0.171 ± 0.100	0.029 ± 0.022	–
Wu *et al.* [63]	2 / 2 22 / 22	0.394 ± 0.154 0.193 ± 0.097	0.064 ± 0.020 0.029 ± 0.014	0.104 0.005
direct-M	17 / 1	0.131 ± 0.077	0.021 ± 0.014	–
isocurves-CRF	2 / 1 17 / 1	0.182 ± 0.118 0.177 ± 0.101	0.027 ± 0.019 0.026 ± 0.017	0.008 0.009
isocurves-MCRF	17 / 1	0.139 ± 0.073	0.022 ± 0.013	0.006

method is able to achieve a better color metric. The cost function used for estimation of M comes from equation (2.19), which allows for the estimation of M to be done by writing equations that only take into consideration color accuracy and do not compare colors at different brightness values. On the one hand, this method is able to benefit from having a cost function that does not favor camera models that better linearize the color space. On the other hand, the method does not linearize the space and, thus, it does not fully describe the camera. Our method does linearize the color space while still obtaining an 20% improvement of chromaticity distance.

To evaluate the repeatability of the method when using general purpose cameras, further tests were performed. Regarding single-channel cameras, to use our method one must assume that the values of ρ^{RAW} are known. While we cannot know in advance the spectrum of the camera and the light, mapping the sRGB albedos to grayscale achieves good repeatability, suggesting that the spectra is not as relevant as in trichromatic cameras with built-in software that perform color mapping. Figure 2.11 shows the median and the inter-quartile range across different poses for the CRF estimation using generic cameras.

As in the previous repeatability tests, the method is able to achieve good repeatability as the inter-quartile range for the CRF estimation is barely noticeable.

2.4.2 Brightness Transfer Functions

For this test, we aim to compare our CRFs estimations to the actual CRF the camera. However, theoretically valid evaluation of the CRF against ground truth is non-trivial under near light, since a direct comparison to ground truth albedos is

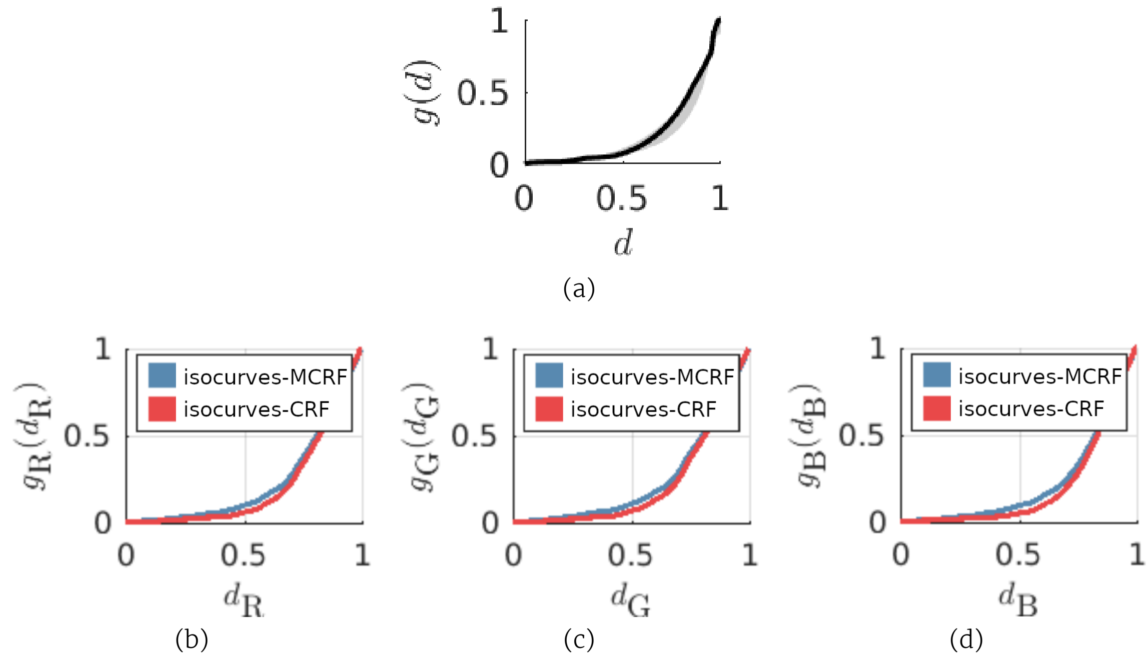


Figure 2.11 – Median and inter-quartile range for the inverse CRFs for generic cameras: (a) monochromatic camera; (b-d) red, green, and blue channels of a cellphone camera. In red is isocurves-CRF method and in blue is the isocurves-MCRF method.

not possible. For a fair comparison we used brightness transfer functions (BTFs) and a single-channel camera so the color mapping and the albedo values do not influence the results. From equation (2.4) we can derive for a monochromatic camera its image formation model as

$$f^{-1}(d(\mathbf{x})) = \alpha v(\mathbf{x})\, q_{\mathbf{x}}(\mathbf{x})\, \rho^{\mathrm{RAW}}(\mathbf{x}) \tag{2.21}$$

where $f : \mathbb{R} \to \mathbb{R}$ is the single channel CRF. The BTF b between two same-pose single-channel images of different exposures, d_1 and d_2, can be described as

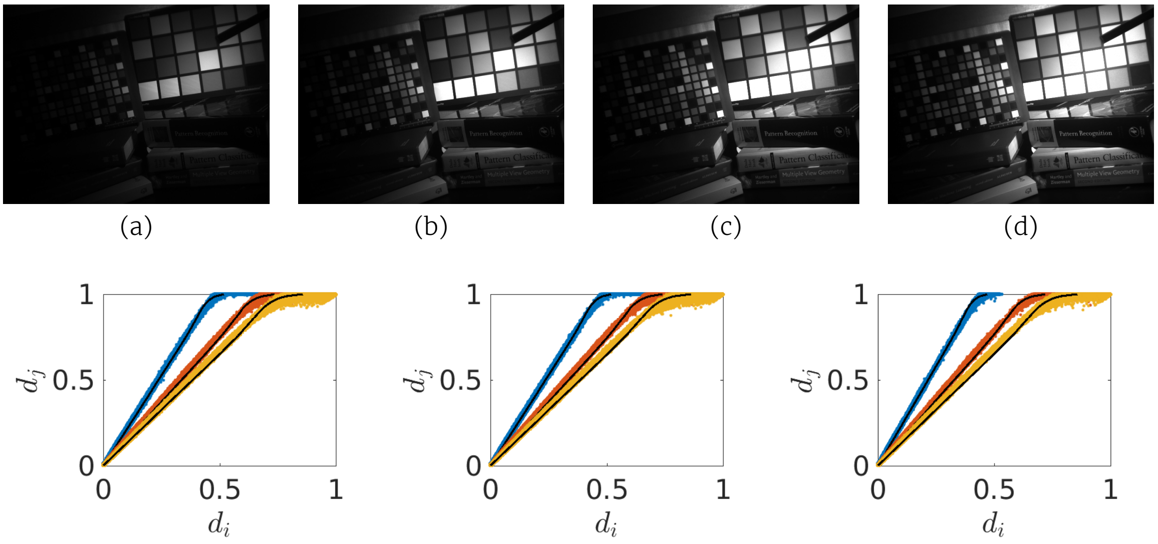
$$d_2(\mathbf{x}) = f^{-1}\left(\frac{\alpha_2}{\alpha_1} f(d_1(\mathbf{x}))\right) = b(d_1(\mathbf{x}))\ . \tag{2.22}$$

By plotting the pixel values of one image against the other one can compare to our estimation of the BTF only by estimating the constant k that minimizes

$$e_{\mathbf{x}} = g(d_2(\mathbf{x})) - kg(d_1(\mathbf{x}))\ . \tag{2.23}$$

Figure 2.12 shows the BTF estimation results for a generic scene imaged with a monochromatic camera with different poses and exposures.

The results show that the estimated BTFs follow the actual BTFs obtained directly



(e) pose 1 (f) pose 2 (g) pose 3

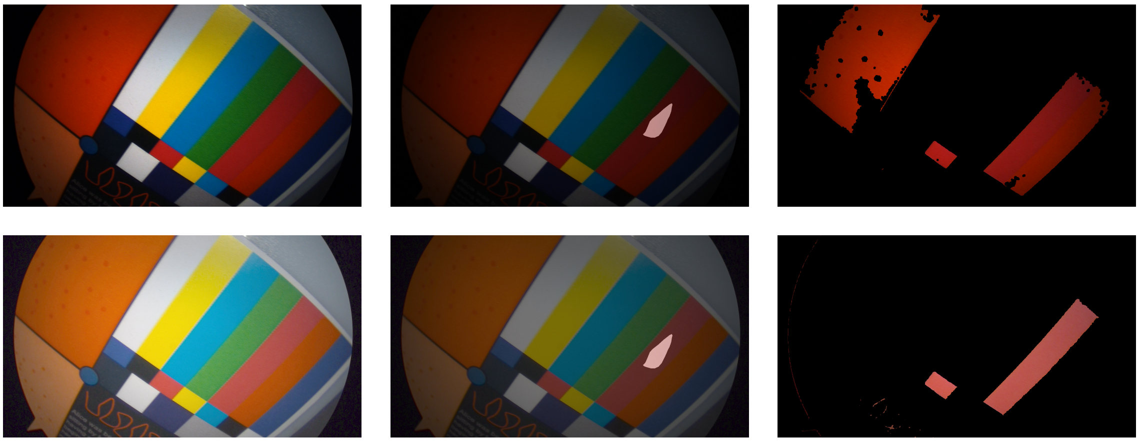
Figure 2.12 – BTF of a monochromatic camera for same-pose images at 4 different exposures. The top row shows the images taken at different exposures for one of the poses. The bottom row presents the ground truth and the estimated brightness transfer functions. Each color represents a pair of exposures, and in black is our estimation.

from the camera. This gives a good indication that the estimated CRF is close to the true camera CRF.

2.4.3 Color Discrimination

Color-based segmentation is used to perform image segmentation based on the color information of the select pixel or group of pixels. Although this type of approach is not usually used in the literature, it can be used here for to demonstrate that camera calibration improves color accuracy and discrimination. With a calibrated camera response, it is expected some improvements on this type of application.

To segment an image based on the chromaticity of a region defined by a user, we implemented a method that uses an hysteresis threshold on the euclidean distance between the chromaticity of the user-defined region and the rest of the image. This method is simple and does aim to compete with the state-of-the-art on color-based segmentation. In contrast, this test aims to show that color-based segmentation can be improved by calibrating the camera. Figure 2.13 shows the chromaticity-based segmentation both before and after calibration of the camera response with the proposed method. It shows a clear improvement in discerning between certain colors, leading to a more accurate segmentation



(a) image (b) hand-drawn segmentation (c) segmented regions

Figure 2.13 – Color-based segmentation with and without camera calibration. The top row is the original image as acquired by the camera and the bottom row is image after correction to sRGB with the proposed calibration method. Note that the segmentation in the top row struggles to differentiate between similar colors, while the image in the second row provides better results.

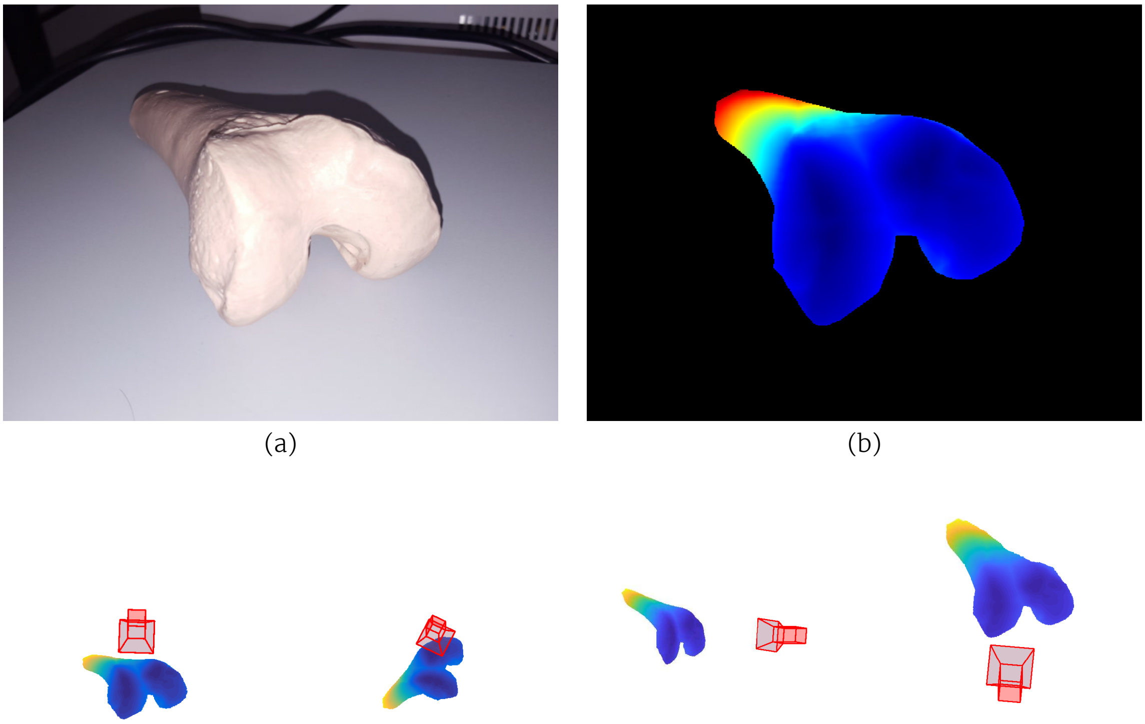
of that particular color defined by the user. These results imply that if this simple method can be improved by calibrating the camera, other methods, such as unsupervised segmentation, can also be improved as long as it uses color as a base for segmentation.

2.5 Applications

Two applications were chosen to showcase the potential of the presented method: SfS using generic cameras and color standardization application for MIP cameras.

2.5.1 Shape-from-Shading

SfS algorithms require linearization of the camera space. To show the applicability of our method, we have performed SfS, using a cellphone camera with its built-in flash light on. We calibrated the camera device using our method and, then, assuming a point light source (not centered in the lens) and the inverse square law for the light falloff, we have used the method made available in [60]. Figure 2.14 shows the results of the reconstruction.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 2.14 – SfS using a calibrated cellphone camera: (a) the original frame, (b) the reconstructed depths, (c–f) 3D reconstructions from different point-of-views.

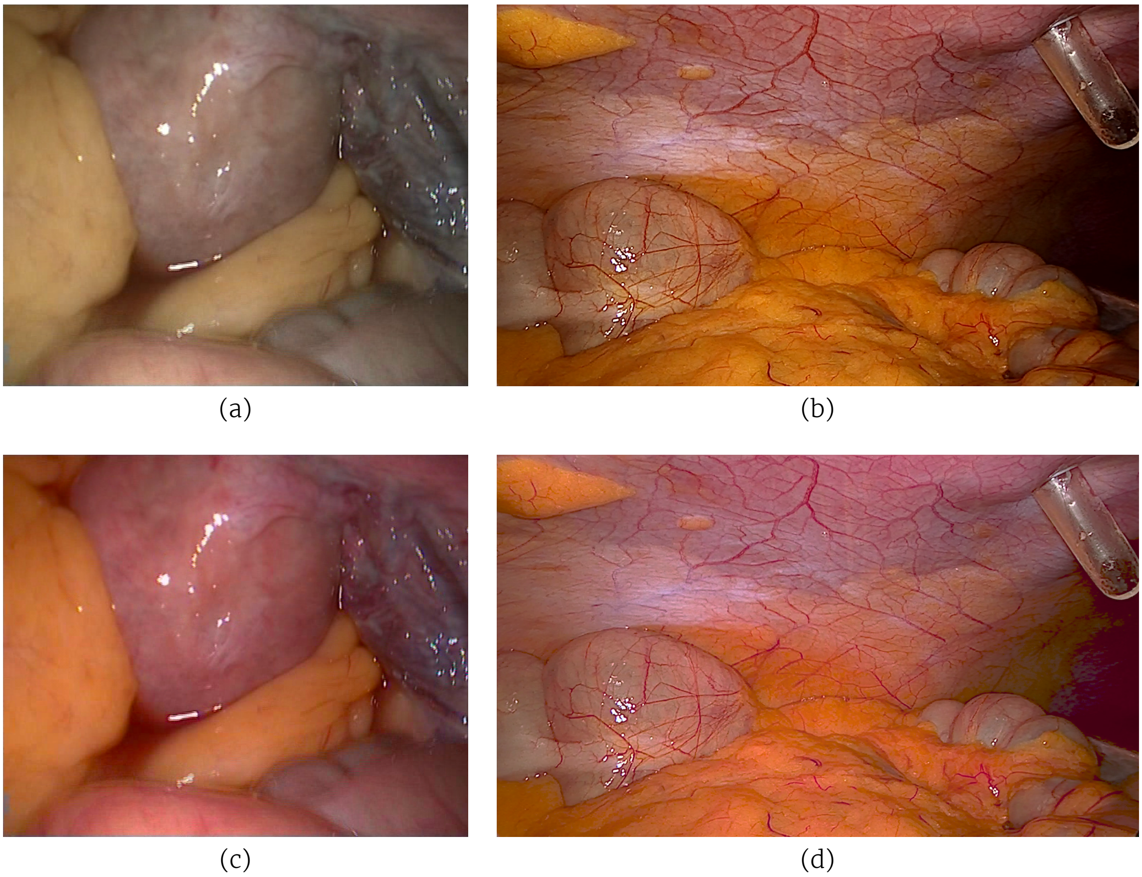


Figure 2.15 – Laparoscopic video frames of similar anatomical cavities from two MIP towers from different vendors: original frames (top), frame after calibration and correction (bottom). Note that the corresponding colors in the bottom row are perceived to be more similar.

2.5.2 Color Standardization

Regarding MIPs, one important application of the camera characterization is the standardization between instruments from different vendors, which inevitably have different camera response properties. Camera characterization is particularly important when post-processing is applied that relies on color information. This is very common in MIP systems. For instance, the visualization toolboxes i-Scan (Pentax Medical, HOYA Corporation, Tokyo, Japan) and SPIES (KARL STORZ GmbH & Co. KG, Tuttlingen, Germany) both use post-processing that manipulates color. Figure 2.15 shows how our method can improve the color standardization between equipments of two different vendors (Storz and Covidien – Medtronic, MN, USA) in abdominal laparoscopy.

The standardization shows noticeable color convergence after the camera response correction, which consequently translates in color convergence in the visualization mode.

2.6 Discussion

In this work, we proposed a camera response characterization method than can be used in cameras with vignetting and/or operating under near light. This single-image method achieved good repeatability results, even better than with Wu *et al.* method [63] that requires 16 images. It was also possible to obtain a 20% improvement on the color accuracy tests. The method is generic since no assumptions are made about the vignetting, and is able to cope with a variety of illuminant types and shapes. Another advantage of our method is that it is easily scalable as more images lead to a new small set of equations that can be stacked if needed. Moreover, if one needs better representation of the sRGB colorspace, adding colors to the targets is the obvious approach. While, for Wu *et al.* approach [63], defining a new color implies a new image, adding colors in our method requires simply adding new regions to the same target.

The proposed method extends a CRF-estimation method to a more complete camera model, while maintaining the single-image requirement. This is crucial as it allows the employment of the method in the OR without disturbing the clinical workflow, or for an easy use in consumer electronics such as cellphone cameras.

As for limitations, our work depends on some assumptions that may not hold in some scenarios. Specifically, the assumption that the illuminants must have the same spectrum and that the ambient light is negligible. While these are limitations to method presented here, they can be guaranteed in the calibration set-up by using a controlled scene where the calibration target is inserted.

3 Color Reproduction and Characterization of Display Monitors

Color gamut is the subset of visible colors that a color output device, such as a display monitor or projector, is able to represent. Two display monitors usually have different color gamuts, due to having different RGB primaries, different ranges of luminance, and different in-monitor color mapping functions. This is a critical point for applications that require color reproducibility on different displays, since each display will represent differently the same image.

One application where color reproduction across displays is crucial is in arrays of display monitors. It requires all the displays to be calibrated to each other to avoid color inconsistencies. Professionals in the fields of digital photography and design also need to perform display characterization and calibration regularly. Color is also an important cue for diagnosis in medical applications where imaging is an integral part of the diagnosis, such as medical endoscopy. If a practitioner is used to a display with a specific color gamut, a change to a completely different display can interfere with diagnosis ability. In fact, display calibration is known to significantly improve practitioner efficiency [3].

Display monitors are usually equipped with a control board that transforms the input discrete logic levels (*i.e.*, the RGB values image or frame to be shown) by a given function. This board has internal memory that stores color parameters which could be changed in a calibration scenario. The display calibration process thus consists in finding the values for the parameters in order to obtain a certain color response (*e.g.*, the BT.709, a recommendation for HDTV from ITU-R — Radiocommunication Sector of the International Telecommunication Union). Currently, this calibration requires precise measuring equipment (*e.g.*, a spectrometer) and a framework that is time consuming for the individual in charge of the calibration procedure. For medical applications, besides the necessary display characterization, display manufacturers often must produce new displays with characteristics similar to the displays already deployed and known

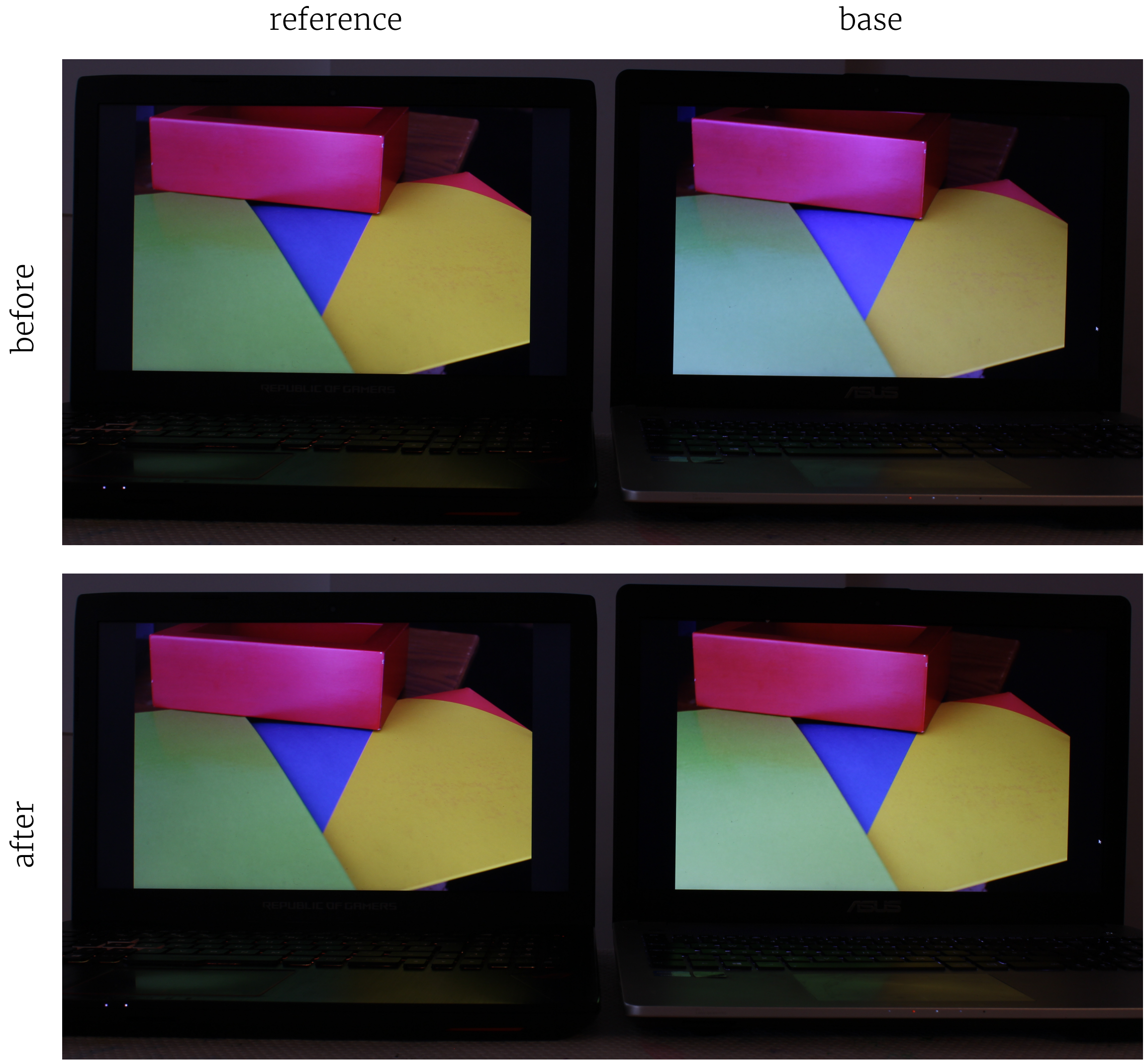


Figure 3.1 – DLSR stills of two displays showing the same input image: before (top) and after (bottom) color reproduction correction of one of the displays.

by the physicians. This leads to a time-consuming trial-and-error process of manual selection of the parameters available to the user, such as brightness, contrast, and temperature.

The main goal of this work is to provide a fast and automatic color reproduction framework for display monitors using a consumer camera. In other words, the same image being displayed in two different displays provides different color output results (see figure 3.1). Therefore, the goal is to estimate which is the best transformation that can be done to the input image of one of the displays so that its' color output matches the other's (see figure 3.2).

Literature has been published on the subject of projector display characterization using specialized measuring equipment (colorimeters, spectroradiometers, spectrometers) [51, 26, 10]. Display monitor characterization using specialized equipment [9] has also been discussed in the literature. However, this special-

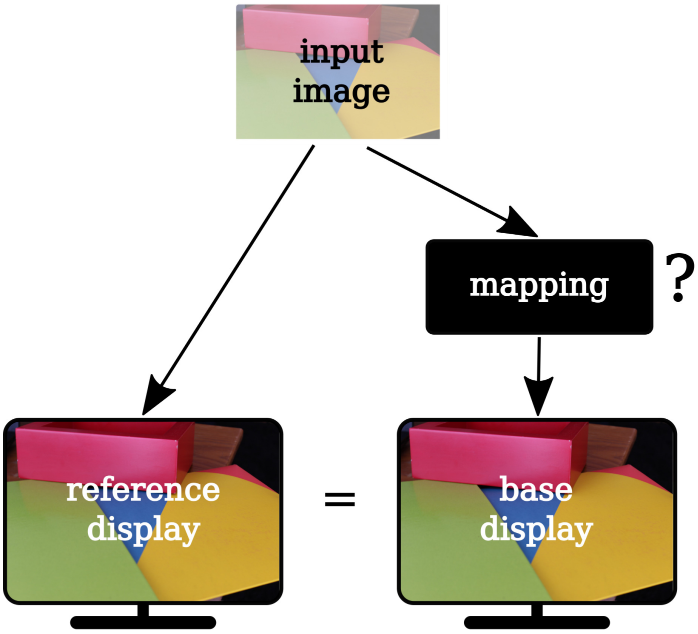


Figure 3.2 – Schematic representation of the goal of the color reproduction framework. The goal is to find which transformation must be done to the image in the base display so that the color output of the base and reference displays is similar.

ized equipment can be expensive and can lead to a tedious and time consuming process for display characterization, where each color must be sequentially shown by the display to be measured individually.

Cameras are known to be able to measure with acceptable accuracy the colorimetric properties of display projectors [4]. Using a camera as a colorimeter has the advantage of being able to take measurements of multiple color patches simultaneously. Therefore, a complete characterization of a display could potentially be produced from a single photograph.

Regarding projector display characterization using cameras, the authors of [4, 54] propose a method that requires the user to visually define the mid-gray level of the display, which introduces subjectivity. Other approaches [7, 6] are targeted to multi-projector arrays where overlap and registration can be used for performing matches between displays and estimating the required mapping functions. Jung *et al.* [29] propose an automatic characterization of display monitors using cameras. However, they used a linear mapping for the calibration and assume that the display monitors are similar in terms of overall luminance. In Post *et al.* [39], the authors aim to calibrate an immutable camera-display system where the display only shows the video streaming of one single camera. That camera is used to calibrate the camera-display system. However, showing images/frames from other camera equipment is not viable.

The main contributions of the present work are a display monitor characterization procedure using a single image taken with a consumer camera and a color reproduction framework for matching the color properties of two or more display monitors that:

- reduces the execution time for display characterization to mere seconds, while still using a comprehensive display model;
- allows for the relaxation of camera acquisition parameters, namely the exposure and the distance to the displays;
- allows for color matching of displays with different overall brightness and different levels of black;
- estimates a parameterized mapping function that outperforms the state-of-the-art methods.

3.1 Overview and Models

To quickly recreate the colors of a reference display monitor in a base display, the method proposed in this work will use a consumer camera (such as a digital single lens reflex (DSLR) camera or a smartphone camera) to take a single photo of each display, which are showing a known calibration image. Fig. 3.3 shows a schematic representation of the process.
An exhaustive description of the method will be provided in section 3.2. First, however, for an easier interpretation of the present work, this section defines the notation used and the main model equations on which this work is based. We discuss the assumptions about the radiometry involved in capturing an image of a display monitor, we describe the display models that are used, as well as the mapping function that is used to transform incoming images on the base display for it to match the reference display. We also propose new images that will be fed to the displays and specify the perceptual metric used in this work.

3.1.1 Camera Measurements

Since we will be using estimation procedures involving minimization of color as measured by the camera, characterization of the camera response must be considered. Photometric calibration of the camera is not mandatory in our algorithms. Modeling the display monitors without a calibrated camera, means that

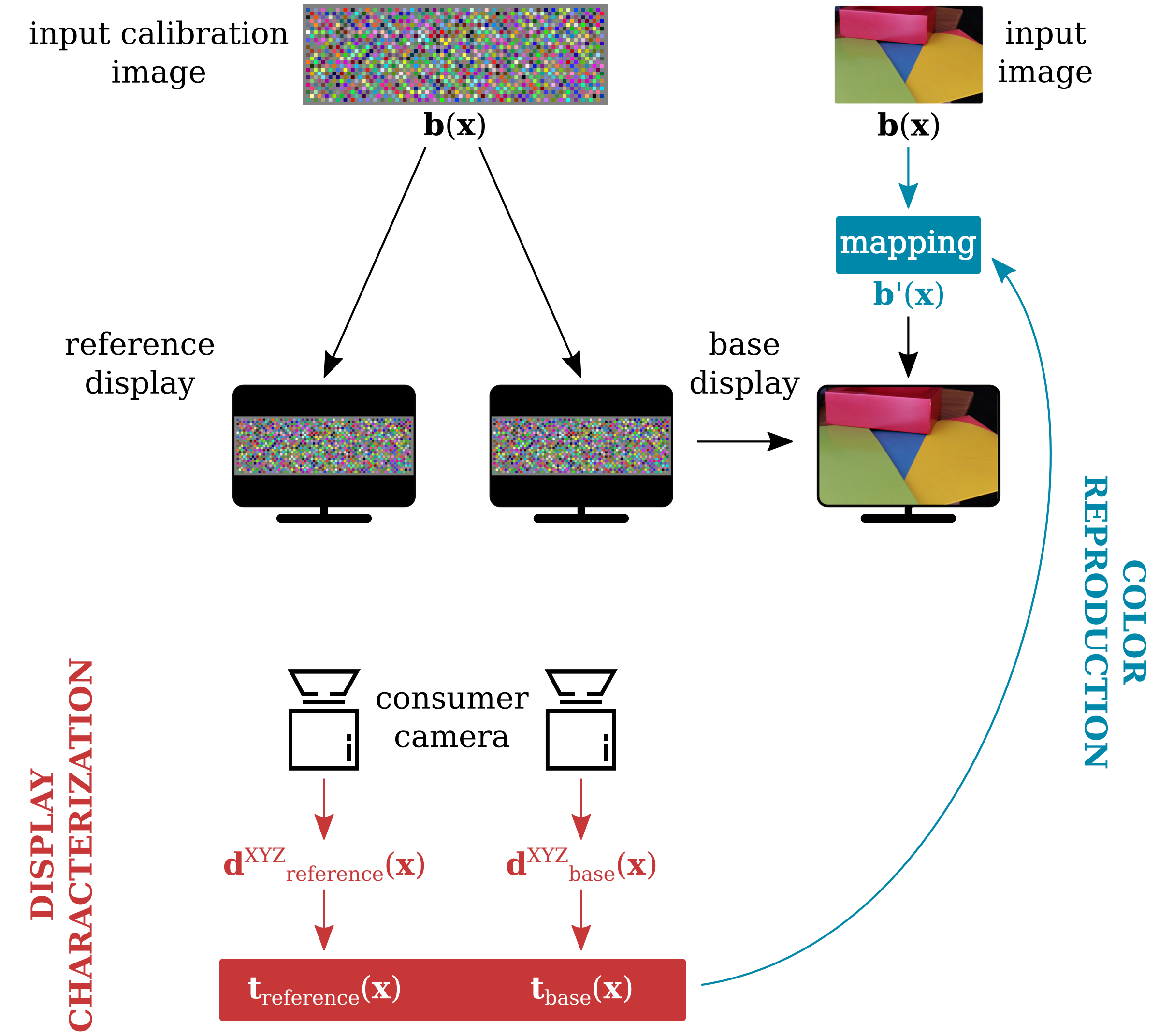

input calibration image
$\mathbf{b}(\mathbf{x})$
input image
$\mathbf{b}(\mathbf{x})$
mapping
$\mathbf{b}'(\mathbf{x})$
reference display
base display
COLOR REPRODUCTION
DISPLAY CHARACTERIZATION
consumer camera
$\mathbf{d}^{XYZ}{}_{\text{reference}}(\mathbf{x})$
$\mathbf{d}^{XYZ}{}_{\text{base}}(\mathbf{x})$
$\mathbf{t}_{\text{reference}}(\mathbf{x})$
$\mathbf{t}_{\text{base}}(\mathbf{x})$

Figure 3.3 – Schematic overview of the display characterization and color reproduction framework.

we will model the display and the camera together. As long as the same camera is used for imaging both the base and the target displays, color reproduction using our method is possible. The minimization space will be warped in relation to visual perception, but this is not crucial to obtain good results.

In this work, we use cameras shooting in RAW mode that are then transformed to a canonical color space, the CIEXYZ, using dcraw [13]. For applications other than color reproduction, the display characterization described in this work is still feasible, however, photometric calibration may be required.

In addition, the cameras that are used do not suffer from significant vignetting and, thus, it will not be considered throughout the rest of the work. Geometric calibration of the cameras is also not considered in this work. It was observed that it was not necessary as long as we use the calibration images that we propose.

3.1.2 Display Monitors

Regarding the display monitors, there are a few assumptions that need to be satisfied experimentally. For imaging the display monitor we assume that there are no external lights and that the monitor radiance is point-wise isotropic. The latter may not reflect the reality for some types of monitors [9], but we minimize this effect by fixating the camera far from the monitor and fronto-parallel to the monitor. It comes that,

$$\mathbf{d}^{\mathrm{XYZ}}(\mathbf{x}) = \alpha \mathbf{l}^{\mathrm{XYZ}}(\mathbf{x}) \tag{3.1}$$

where $\mathbf{d}^{\mathrm{XYZ}}$ is the acquired image in the CIEXYZ space, $\mathbf{x}$ is a scene point (in this case a point on the monitor), $\mathbf{l}^{\mathrm{XYZ}}$ is the radiance and α is the camera exposure. For simplicity, the equation is written in the CIEXYZ color space, but it could easily be extended to use other relevant color spaces.

Given the mentioned assumptions, the radiance can be then measured up-to-scale directly by a camera. Additionally, modeling of a display monitor can be done directly from the values measured by the camera.

In this work, we used a 3D LUT as a model for the display monitors. Note that for a complete 3D LUT with 8-bit axis one would need 256^3 measurements. This is not feasible. Nevertheless, accurate representation of the display can still be achieved with a minimum of 1000 measurements [54].

The camera has a striking advantage over the colorimeter/spectrometer, as it can perform all 1000 measurements in a single image. This approach is impractical with other, more standard, measurement equipment, where only a single color can be measured at a time. The 3D LUT is built directly from a single acquired image $\mathbf{d}$ as a series of measurements, for instance 1000 different colors spread out through the RGB space as explained in section 3.1.4. The remaining values are retrieved by interpolation. We formalize the model as

$$\mathbf{d}^{\mathrm{XYZ}}(\mathbf{x}) = \mathbf{t}(\mathbf{b}(\mathbf{x})) . \tag{3.2}$$

where $\mathbf{b}$ is the input image, *i.e.*, the image fed to the display, and $\mathbf{t}$ is the 3D LUT, an $\mathbb{R}^3 \to \mathbb{R}^3$ function.

Other models were considered, such as the PLCC* and the PLCC model [54]. However, better results were obtained with the LUTs.

3.1.3 In-monitor Mapping Function

Not all operations are permitted for color correction in display firmware and/or software dedicated for display. For instance, there is no 3D LUT in most display monitors firmware. There are only a few specific operations that can be performed. In this work, it is assumed that the allowed operations are a matrix multiplication, $\mathsf{R}_{3\times3}$, followed by a function $\mathbf{h}$, composed of three $\mathbb{R} \to \mathbb{R}$ functions, one for each channel. Thus, to transform the input image $\mathbf{b}$ to a new image $\mathbf{b}'$ that compensates for the differences between two monitors, we have

$$\mathbf{b}' = \mathbf{h}(\mathsf{R}\mathbf{b}) . \tag{3.3}$$

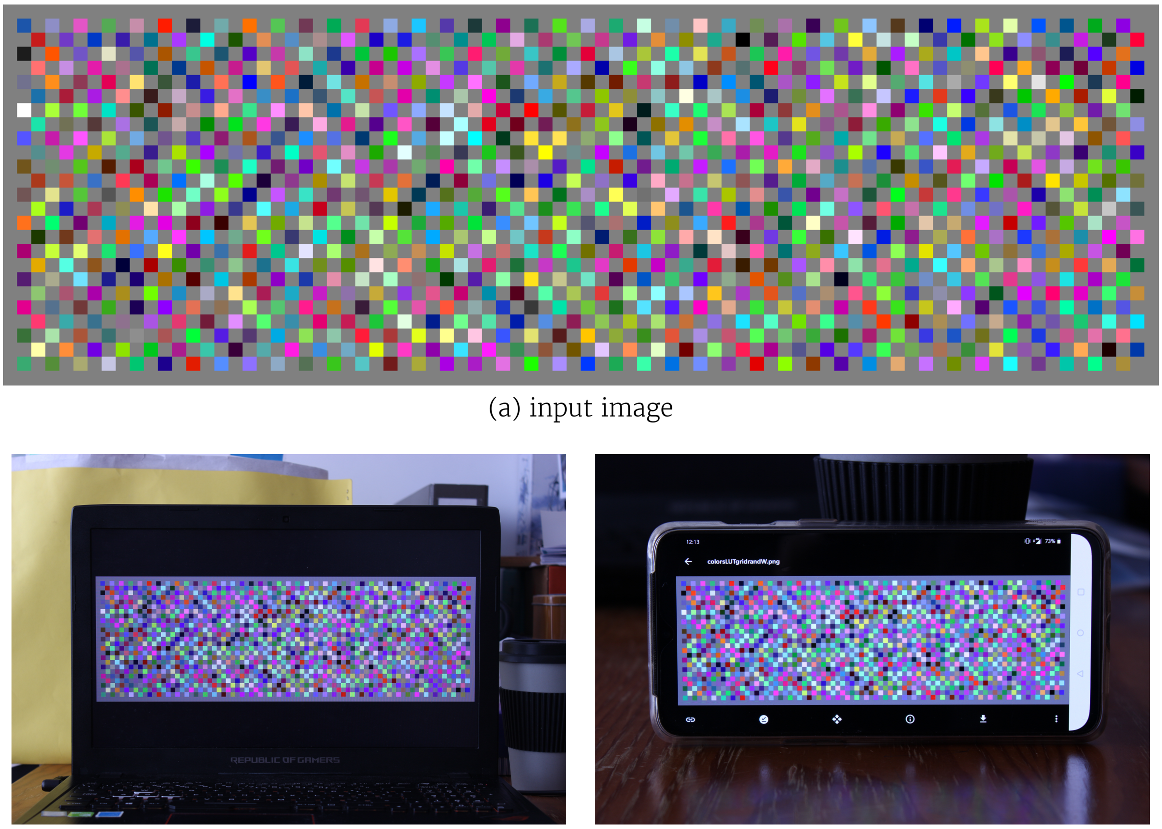
In this instance, we have used 4th-order polynomials for function $\mathbf{h}$, as it presented a good trade-off between complexity and results.

Although this is not a standard transformation, this is can be used in many display manufacturers as it is composed by traditional color operations and gamma curve manipulations. Our method, however, is not closed off to other mapping functions. In fact, we have also tested another mapping where only the matrix multiplication is used, as in [29]. Formally,

$$\mathbf{b}' = \mathsf{R}\mathbf{b}. \tag{3.4}$$

3.1.4 Input Calibration Image

The input image used for display characterization is shown in figure 3.4. The image has 1000 color patches that correspond to the values on a 10 by 10 by



(a) input image

(b) reference display

(c) base display

Figure 3.4 – Image fed to the display monitors for display characterization: (a) 1000-color image used for modeling the display models and to estimate the parameters for color reproduction mapping; (b) example of an image acquired with a DSLR camera of a reference display showing the characterization image; (c) example of an image acquired with a DSLR camera of a base display showing the characterization image.

10 grid in the RGB color space. Each patch is surrounded by gray patches to reduce spatial overlap between colors due to camera blur and/or defocus, and to attenuate spatial color variation present in some display technologies. For instance, a pure black patch surrounded by red pixels can have a different reading than one surrounded by green pixels.

Another potential source of error is when there is no consistency across the entire display. For example, due to a non-uniform backlight in LCDs. This could lead to changes in the measured colors in some regions of the display. The gray patches could also be used to normalize for this aspect. However, in our tests, as long as the camera is fronto-parallel and far from the display monitors, this operation was not necessary. Positioning the camera in such way also ensures that, from the point view of the camera, violations in the assumption that the display radiance is point-wise isotropic, are less noticeable.

3.1.5 Perceptual Metric

The $u'v'$ chromaticity plane provides a good two-coordinate color description. It can be directly transformed from the CIEXYZ color space. Within this color plane, the perceived difference between two colors can be expressed as an euclidean distance, $\Delta_{u'v'}(\mathbf{d}_1, \mathbf{d}_2)$. This metric can effectively be used as a color distance metric [12], and will be used to quantitatively evaluate the models and to perform display matching.

3.2 Color Reproduction

To achieve color reproduction we combine the concepts disclosed in the previous sections. For an overview of the complete process, please check algorithm 3.1.

Before any optimization, the single-frame measurements acquired with the camera must be transformed into a LUT. The LUT is populated with the median pixel values taken from the corresponding color patches shown on screen. This is done for both the base and the reference displays. Since we are using the 1000-color image proposed in section 3.1.4, we end up with LUTs that have $10 \times 10 \times 10$ triplets of XYZ values. In this work, we use linear interpolation to obtain the remaining values since more complex interpolation methods did not seem to provide better results.

Algorithm 3.1 Display monitor color reproduction algorithm.

```
Require: look-up table for reference display, $\mathbf{t}_{\text{reference}}$
  obtain look-up table for base display, $\mathbf{t}_{\text{base}}$
  estimate $\kappa_1$ and $\kappa_2$ — equation (3.8)
  estimate $\mathsf{R}$ and $\mathbf{h}^{-1}$ — equation (3.10)
  estimate $\mathbf{h}$ — equation (3.10)
  non-linear refinement of $\mathsf{R}$ and $\mathbf{h}$ — equation (3.11)
```

The goal of the color reproduction procedure is to find what new image $\mathbf{b}'(\mathbf{x})$ must be given to the base display so that the measured LUTs of both the base and the reference displays are equivalent. For simplicity, let us first define $\mathbf{b}_\rho$ and $\mathbf{b}'_\rho$ as a shorthand for the colors being displayed in patch ρ of, respectively, the images $\mathbf{b}(\mathbf{x})$ and $\mathbf{b}'(\mathbf{x})$. Formally,

$$\mathbf{b}_\rho = \mathbf{b}(\mathbf{x}_i), \quad \forall i \in \mathcal{P}_\rho$$

$$\mathbf{b}'_\rho = \mathbf{b}'(\mathbf{x}_i), \quad \forall i \in \mathcal{P}_\rho$$

where $\mathcal{P}_\rho$ is the set of pixel indexes that correspond to a color patch ρ. Then,

$$\mathbf{t}_{\text{base}}\left(\mathbf{b}'_\rho\right) \equiv \mathbf{t}_{\text{reference}}\left(\mathbf{b}_\rho\right), \quad \forall \rho \tag{3.5}$$

where $\mathbf{t}_{\text{base}}$ and $\mathbf{t}_{\text{reference}}$ are, respectively, the measured LUTs of the base and the reference displays. In fact, we must know, not only $\mathbf{b}'$, but how $\mathbf{b}'$ can be obtained from $\mathbf{b}$. Combining (3.3) and (3.5) we have that

$$\mathbf{t}_{\text{base}}\left(\mathbf{h}\left(\mathsf{R}\mathbf{b}_\rho\right)\right) \equiv \mathbf{t}_{\text{reference}}\left(\mathbf{b}_\rho\right), \quad \forall \rho. \tag{3.6}$$

The unknowns are the in-monitor mapping function, composed by the matrix R and the polynomials $\mathbf{h}$. However, we still need to define how the comparison between the two displays must be performed.

There are multiple factors that must be considered when comparing two displays. Among the factors that make this comparison non-trivial is the variable camera exposure. The gamuts of the displays are also important in this comparison. Not only the color gamut defined by the 3 primary colors of the display but also the full 3D gamut of the display. The gamut of display is in reality defined, not only by its primary colors, but also by the brightness range for each color that can be represented. Different displays can have very different overall brightness values (*e.g.*, cellphones and some medical displays have more brightness than other traditional displays). How black is the pure black of a display is another factor that must be taken into account (*e.g.*, OLED displays can achieve darker levels of black than other types of displays). The full 3D gamuts must be considered in the comparison, because the in-monitor mapping function that we need to estimate should not change the overall brightness of the base display or how dark the black level is, which would happen if a direct comparison of the two display were to be used. This is not desirable because, in the one hand, brighter displays should not lose their brightness when being matched to a reference with less brightness, on the other hand, a base display with less overall brightness cannot be matched directly to a brighter reference. Similarly, a darker pure black is a desirable characteristic that should not be changed to accommodate a reference with a brighter black level. Additionally, camera exposure and distance of the camera to the display can also have an effect similar to differences in brightness between the displays. The present work aims to achieve display monitor color reproduction without the need to have static camera exposure or to have similar displays being photographed at the same distance. This allows for color reproduction of a wide variety of displays and relaxation of acquisition settings. All these factors haven been taken into consideration when defining how to compare the LUTs of two displays.

Nonlinear optimization is used to estimate the in-monitor mapping function. In this way, we are able to perform the optimization with a cost function related to the human perception of color differences. However a good initialization is required for good results and fast optimization. To perform the comparison between the two LUTs in both the initialization and nonlinear refinement stages, two unknown scalars were introduced, a scaling and a shift. These scalars will compensate for the differences in the camera exposures, in the distances of the camera to the display, in the brightness of the pure black of the displays, and in the overall brightness.

3.2.1 Initialization

The initialization is performed in three optimization steps.

The first step is to match the codomains of the two LUTs. To compare the two LUTs in the initialization stage we define

$$\mathbf{t}_{\text{base}}\left(.\right) = \kappa_1 + \kappa_2 \mathbf{t}_{\text{reference}}\left(.\right) \tag{3.7}$$

where κ_1 and κ_2 are unknown scalars, the aforementioned scalars for scaling and shift. These scalars are estimated using L^2-norm with equations of the form

$$t_{\text{Y,base}}\left(.\right) = \kappa_1 + \kappa_2 t_{\text{Y,reference}}\left(.\right) \tag{3.8}$$

where t_{Y}, an $\mathbb{R}^3 \to \mathbb{R}$ function, is the Y channel of the LUT $\mathbf{t}$. The reasoning for using the channel Y instead of using a quantity that better relates to human perception, such as lightness L^* (CIELUV), is that this change would require the definition of a white point, which is not crucial for our approach. The values that are outside the common codomain are ignored for the rest of the initialization.

From (3.6) and (3.7), we can write

$$\mathbf{h}\left(\mathsf{R}\mathbf{b}_\rho\right) = \mathbf{t}^{-1}_{\text{base}}\left(\kappa_1 + \kappa_2 \mathbf{t}_{\text{reference}}\left(\mathbf{b}_\rho\right)\right), \quad \forall \rho. \tag{3.9}$$

At this point the right hand side of the equation is fully known/initialized. It will be denoted as $\mathbf{b}'^{*}_{\rho}$, as this could be used for a first approximation of $\mathbf{b}'\left(\mathbf{x}\right)$.

Thus,

$$\mathbf{h}\left(\mathsf{R}\mathbf{b}_{\rho}\right) = \mathbf{b}'^{*}_{\rho}, \quad \forall \rho. \tag{3.10}$$

For the second step, the set of equations defined in (3.10) can used to estimate R and the inverse of **h**, one channel at a time, without additional unknowns. This convex optimization problem was performed with quadratic programming and linear inequality constraints on the monotonicity of the polynomials.

Finally, in the third step, the direct **h** is estimated using the matrix R that was estimated in second step. This estimation of **h** is necessary since the polynomials used for $\mathbf{h}^{-1}$ are not invertible. In this way, we are able to estimate R without assuming a linear **h**, which in turn gives a more accurate initialization for both R and **h**. The estimation in the third step is also performed for each channel independently, using quadratic programming and linear inequality constraints on the monotonicity of the polynomials.

3.2.2 Nonlinear Refinement

For the nonlinear optimization the cost function we will be composed of two metrics: the aforementioned perceptual metric and a metric comparing the brightness of the color patches. This second metric ensures that the relationships between the color patches are maintained. Without it, the brightness of the colors could lose its meaning.

The comparison used here to match the codomains of the two LUTs takes the form of (3.8). It still uses a shift κ_1 and scaling κ_2 in the Y channel. However, κ_1 and κ_2 are not refined for. They are taken as constants for the nonlinear refinement.

The optimization problem can be written as

$$\min_{\mathbf{h}(.),\mathsf{R}} \frac{1}{N_{\rho}} \sum_{\rho} \epsilon_{\rho}^{\text{chromaticity}} \left(\mathbf{h}\left(.\right), \mathsf{R}\right) + \lambda \frac{1}{N_{\rho}} \sum_{\rho} \epsilon_{\rho}^{\text{brightness}} \left(\mathbf{h}\left(.\right), \mathsf{R}\right) \tag{3.11}$$

with

$$\epsilon_{\rho}^{\text{chromaticity}} \left(\mathbf{h}\left(.\right), \mathsf{R}\right) = \Delta_{u'v'} \left(\mathbf{t}_{\text{base}} \left(\mathbf{h}\left(\mathsf{R}\mathbf{b}_{\rho}\right)\right), \mathbf{t}_{\text{reference}} \left(\mathbf{b}_{\rho}\right)\right) \tag{3.12}$$

and

$$\epsilon_{\rho}^{\text{brightness}} \left(\mathbf{h}\left(.\right), \mathsf{R}\right) = \left|\kappa_1 + \kappa_2 t_{\text{Y,reference}} \left(\mathbf{h}\left(\mathsf{R}\mathbf{b}_{\rho}\right)\right) - t_{\text{Y,base}} \left(\mathbf{b}_{\rho}\right)\right|. \tag{3.13}$$

Within the cost function, some values of transformed image $\mathbf{b}'(\mathbf{x})$ (see (3.3)) may be outside the range of possible values. We used absolute colorimetric rendering

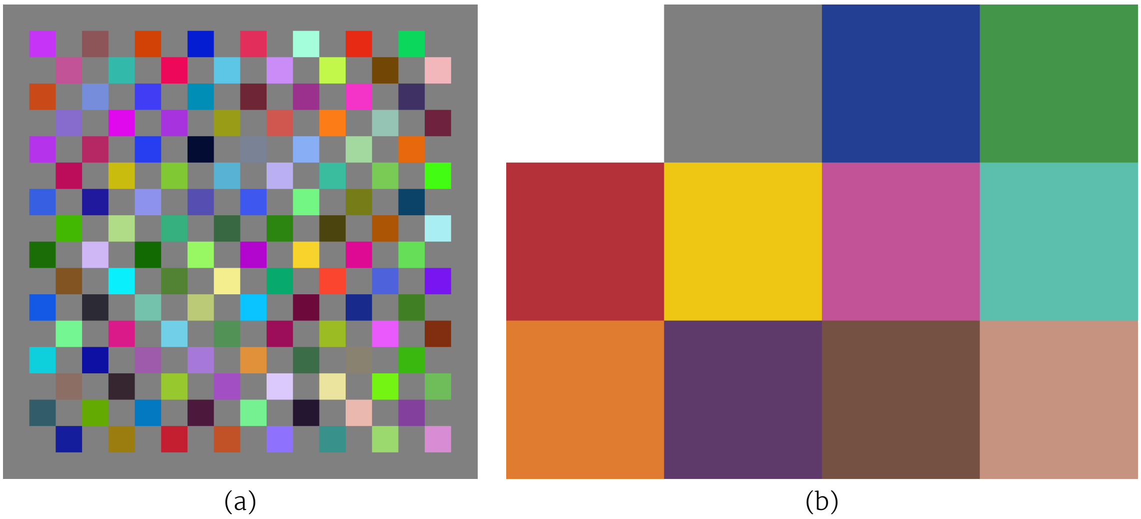


Figure 3.5 – Images fed to the display monitors: (a) the image with 128 randomly-chosen colors that was used for quantitative assessment of the models and the mapping; (b) the image with 12 colors used for qualitative visual assessment.

intents to bring them back into the cube of possible values.

Note that κ_1 and κ_2 are only used to compare the two LUTs. They will not be used after the optimization is performed. Only R and $\mathbf{h}$ are needed.

3.3 Experimental Validation

In this section, we will discuss the evaluation procedures and their results.

The cameras used in this work were: (**C1**) a digital single-lens reflex Canon EOS 600D (Canon Inc., Tokyo, Japan) with a standard Canon EF-S 18-55mm F3.6-5.6 IS II lens, and; (**C2**) a smartphone camera, the OnePlus 7 (OnePlus Technology Co., Ltd., Shenzhen, China) built-in camera.

The monitors used for evaluation are: (**D1**) an ASUS N56VZ (ASUSTeK Computer Inc., Taipei, Taiwan) laptop display; (**D2**) an ASUS ROG GL553V laptop display; (**D3**) an OnePlus 7 OLED display; and (**D4**) the same ASUS ROG GL553V display with manual color settings. For experimental validation, in addition to the 1000-color image (see figure 3.4), two additional images were displayed in each monitor and photographed by the camera. One image with 128 randomly generated colors (quantitative assessment), and another with 12 hand-picked colors to match the colors used in the X-Rite (X-Rite, Inc., MI) ColorChart (qualitative assessment). See figure 3.5.

A few values for the optimization parameter λ were tested (see equation (3.11)).

Table 3.1 – Metrics of chromaticity distance for the display color reproduction frameworks. Initial and final $\Delta_{u'v'}$ (average±standard deviation). Each line corresponds to a different display monitor pair. The lines are sorted by the perceptual metric obtained before corrections.

base	reference	before	Jung *et al.*	ours (R only)	ours
D3	D2	0.028 ± 0.025	0.017 ± 0.019	0.013 ± 0.016	$\mathbf{0.012 \pm 0.016}$
D2	D3	0.028 ± 0.025	0.015 ± 0.013	$\mathbf{0.013 \pm 0.011}$	$\mathbf{0.013 \pm 0.011}$
D2	D1	0.039 ± 0.027	0.025 ± 0.028	0.018 ± 0.017	$\mathbf{0.017 \pm 0.016}$
D3	D4	0.045 ± 0.028	0.022 ± 0.018	0.016 ± 0.014	$\mathbf{0.010 \pm 0.009}$
D2	D4	0.051 ± 0.039	0.032 ± 0.029	0.017 ± 0.013	$\mathbf{0.008 \pm 0.006}$
D3	D1	0.056 ± 0.036	0.024 ± 0.019	0.020 ± 0.017	$\mathbf{0.019 \pm 0.016}$
D1	D4	0.057 ± 0.050	0.027 ± 0.024	0.018 ± 0.015	$\mathbf{0.011 \pm 0.009}$

The best results were obtained with $\lambda = 1$. Regarding the in-monitor mapping function, 4th-order polynomials were used for function $\mathbf{h}$.

3.3.1 Color Reproduction

For evaluating the color reproduction framework we resort to the 128-color input calibration image. One shot of the display is taken with the camera before correction and another after correction using the parameters R and $\mathbf{h}$. For these results, only the DSLR camera (C1) was used. Table 3.1 shows the results. For comparison, we implemented a version of our approach where only a 3×3 matrix R is estimated as the in-monitor mapping function, as in (3.4), instead of both R and $\mathbf{h}$. We also provide baseline results using our implementation of the method proposed by Jung *et al.* [29], which also estimates only a 3×3 matrix.

The baseline method reduces the color perception metric by an average of 46%. Our approach achieves better metrics for all tested cases and is able to achieve an average reduction of 68% and up to 84%.

The simpler version of our approach (estimation of only R) is able to outperform the baseline method for all cases. These improvements are due to the fact that we use a cost function based in color perception and to the better codomain matching, *i.e.* 3D gamut matching, between the two displays. By using the 3D LUTs and by using scaling and shift parameteres in the brightness channel, we are able to achieve better results, even when estimating the same transformation (only R).

Figures 3.6, 3.7, and 3.8 show the optimization results for the different display monitor pairs. Figures 3.6 and 3.7 presents the distances of the colors, in the chromaticity plane, of some pairs of base-reference display monitors.

In this figure, one evaluate how the distances between the colors of the base and the reference displays are improved with our framework. In addition, the chromaticity diagram is shown to allow for visual assessment of what the color distances in the $u'v'$ plane represent in terms of visual perception of color distances.

In figure 3.8, one can visually assess the results of the calibration. Note that it is not expected that the colors are matched in terms of brightness as the display monitors might have different overall brightness. Only the tone of the color is supposed to match. The 4 colors in the bottom right corner of the base displays are perceptually closer to the reference. Only one color patch, the white patch of display D3, seems to not be completely corrected. This is expected, because we are estimating an in-monitor mapping function that must be implemented in real applications. It is not a perfect transformation that is able to map correctly all possible colors. Nevertheless, all other color patches seem to be perceptually closer to the reference image.

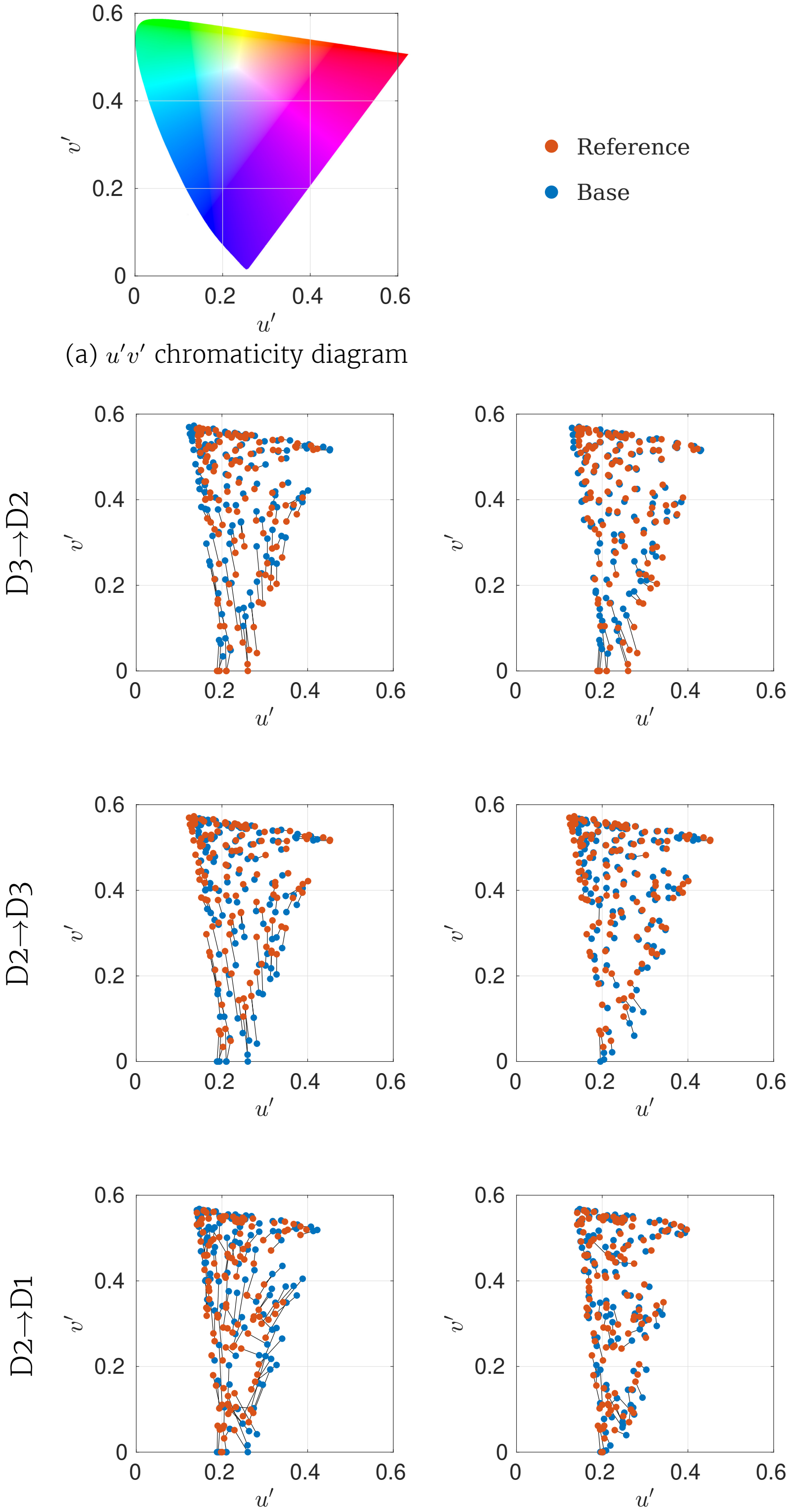
3.3.2 Colorimeter Comparison

In this experiment we want to show if a calibration with our method is able to match that of a hardware specifically designed for display monitor calibration. For that purpose, we used a Datacolor Spyder 3 Elite (Datacolor, NJ), which is a colorimeter for display calibration usually used in professional setups, such as professional digital photography, where color accuracy is very important.

The colorimeter was used to calibrate one of the display monitors to a standard color space by using an unknown mapping function. We then tried to match the mapping performed by the colorimeter with our camera-based method, by using camera shots of the display showing our input calibration image before (base) and after (reference) the colorimeter calibration. A perceptual metric of zero would mean that the colorimeter and our method lead to the same result. Nevertheless, small differences are expected due to measurement errors and differences in the mapping function.

The results are shown in table 3.2. For these results, only the DSLR camera (C1) was used.

The results also show the potential of using this method to calibrate display monitors to standard color spaces, such as the BT.709, without having photometrically calibrated cameras. Since, in that case, a camera shot of the reference display already calibrated to the standard color space would be necessary.

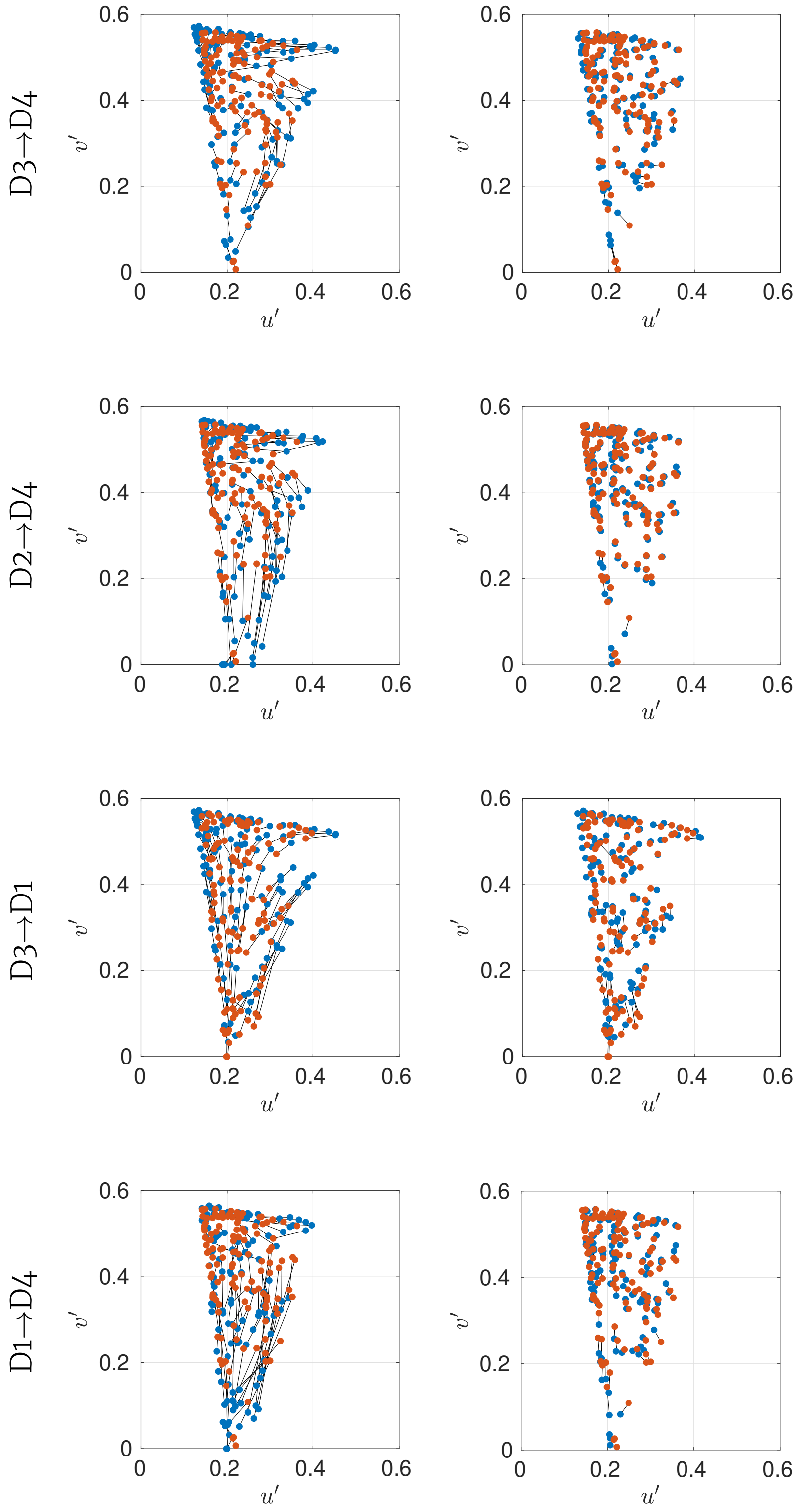


(a) $u'v'$ chromaticity diagram

(b) before correction

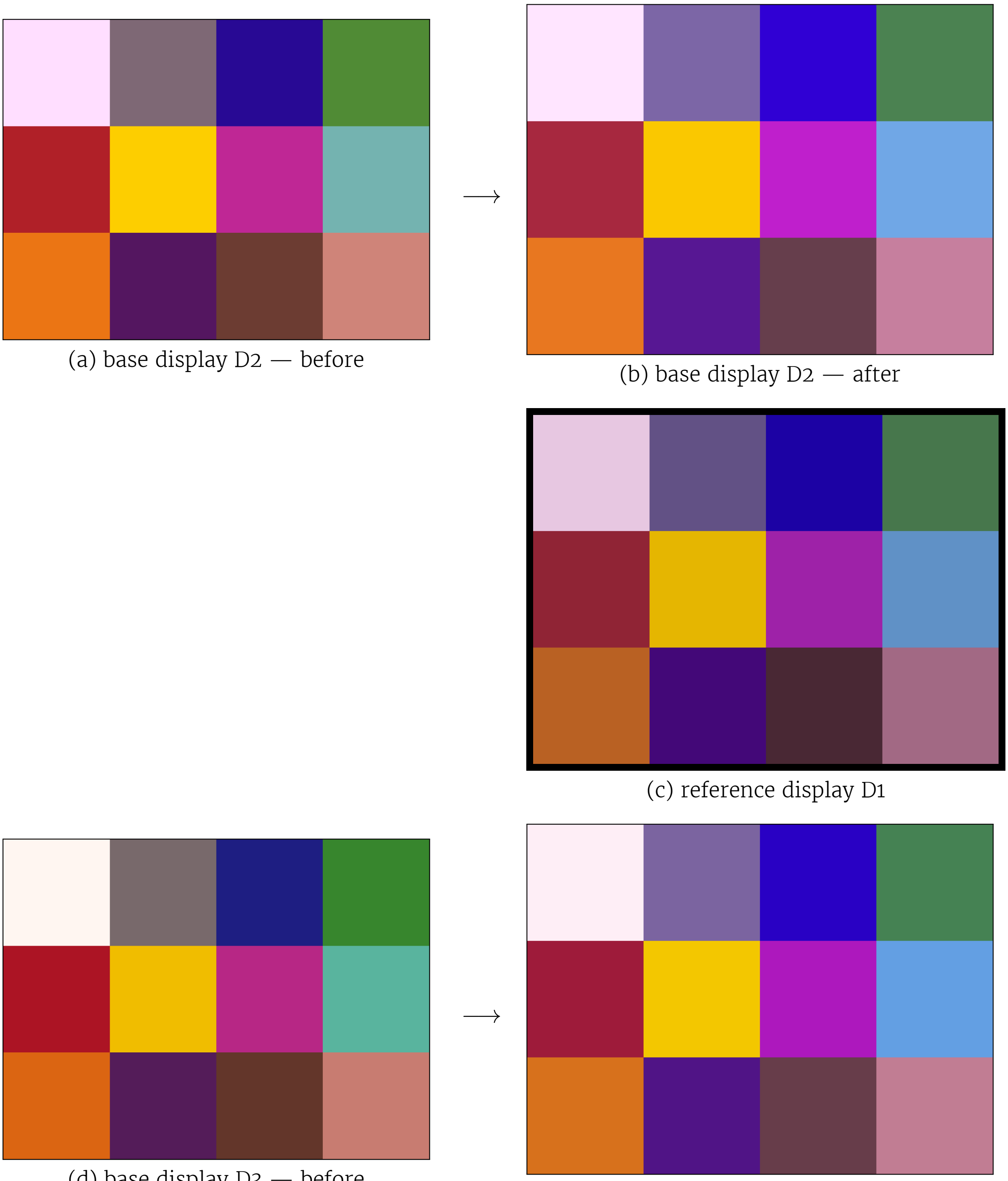
(c) after correction

Figure 3.6 – Color distances before and after the application of the color reproduction framework as represented in the $u'v'$ chromaticity plane. Plot (a) shows the CIE 1976 UCS (uniform chromaticity scale) diagram, *i.e.*, the $u'v'$ diagram. The distances were taken with the DSLR camera and are relative to the 128-color validation image. The red dots represent colors in the reference display and the blue dots represent colors in the base display.



(a) before correction (b) after correction

Figure 3.7 – Color distances before and after the application of the color reproduction framework as represented in the $u'v'$ chromaticity plane (continuation from figure 3.6). The distances were taken with the DSLR camera and are relative to the 128-color validation image. The red dots represent colors in the reference display and the blue dots represent colors in the base display.



(a) base display D2 — before

(b) base display D2 — after

(c) reference display D1

(d) base display D3 — before

(e) base display D3 — after

Figure 3.8 – Color patches shown in the display monitor and imaged with the DSLR camera. The patches are shown here in the sRGB color space for qualitative assessment of the improvements obtained with the color reproduction framework. Each column relates to a different display monitor. The first and the third columns relate to the base displays that were calibrated to match the reference display represented in the second column. Note that the 4 colors in the bottom right corner of the base displays are perceptually closer to the reference display.

Table 3.2 – Display color reproduction metrics when matching a colorimeter with a camera-based method. Initial and final $\Delta_{u'v'}$ (average±standard deviation) for the color reproduction frameworks. For this test, the reproduction frameworks must use a monitor calibrated with a colorimeter as a reference display and the same display before correction as the base display.

before	Jung *et al.*	ours (R only)	ours
0.053 ± 0.037	0.015 ± 0.013	0.012 ± 0.012	$\mathbf{0.007 \pm 0.008}$

Table 3.3 – Cross-validation results of the color reproduction framework using multiple cameras. Initial and final $\Delta_{u'v'}$ (average±standard deviation) for the proposed color reproduction framework.

acquisition camera	before	validation with C1	validation with C2
C1	0.051 ± 0.039	0.007 ± 0.007	0.008 ± 0.008
C2	0.046 ± 0.035	0.007 ± 0.006	0.007 ± 0.006

3.3.3 Camera Cross-Validation

For the final experiment, we aim to demonstrate the robustness of the method across different cameras. Since the cameras are not previously calibrated photometrically, results will differ. To assess this problem we performed cross validation using both cameras, a DSLR camera (C1) and a smartphone camera (C2). The displays used in this test were: D2 as base display and D4 as reference display.

The results are shown in table 3.3 and confirm that photometric calibration of the cameras is not necessary to obtain robust results.

3.4 Discussion

We have provided a framework for color reproduction across display monitors using a single image taken with a common camera, such as a smartphone camera. This work is relevant in many different applications. In a medical context, color reproduction is crucial as color information acquired from a video camera will be seen by surgeons and other physicians through many monitors, both in and out of the OR. In more generic applications, digital photography displays and multi-display arrays can also be calibrated with the proposed method.

Our method was able to achieve better results than other camera-based methods (Jung *et al.* [29]) for this type of problem, and it was shown to be robust across cameras even without photometric calibration. Additionally, our method can be easily extended to other in-monitor mapping function, as long as initialization of those mapping functions is feasible.

4 6-D Pose Estimation for Augmented Reality

Osteoarthritis is a joint disease that causes pain and stiffness due to damage of the joint cartilage and the underlying bone [50]. It is the most common joint disease in the world, with an estimated prevalence of 14% in adults with 25 years or older and 34% with 65 years or older [50]. Depending on the severity of the symptoms, the treatment options can vary from non-operative to a joint arthroplasty. Total knee arthroplasty (TKA) is the principal choice for improving the quality of life of patients suffering from advanced knee arthritis. It is estimated that the demand for TKA in the United States will approach 3.5 million cases per year by 2030 [31]. Although being one of the most effective surgical options to reduce pain and restore the knee function, about 20% of the patients undergoing a TKA surgery are not satisfied [8]. As discussed in [55], there are important surgical variables (*e.g.*, lower leg alignment and soft tissue balancing) that have a direct impact in the success of TKA, which are manually controlled by the orthopedic surgeon. It requires experience for accurately combining all these surgical variables into an optimal implant alignment. To assist the surgeon in controlling these variables, several computer navigation systems have been developed [55].

Existing TKA navigation systems require a sensing technology for performing anatomical measurements or supporting the surgeon in following a preoperative plan of the bone resections. The most widely used technology for this purpose is optical tracking (*e.g.*, NAVIO Surgical System — Smith & Nephew plc, Watford, United Kingdom). While providing accurate 3D measurements, navigation systems based on optical tracking have three main drawbacks: (1) optical tracking platforms are costly, which is one of the reasons for the high cost of existing navigation systems, (2) they necessitate the insertion of pins in the distal femur and proximal tibia for fixing the markers to be tracked, requiring additional bone incisions and surgical time, and (3) the trackers to be attached

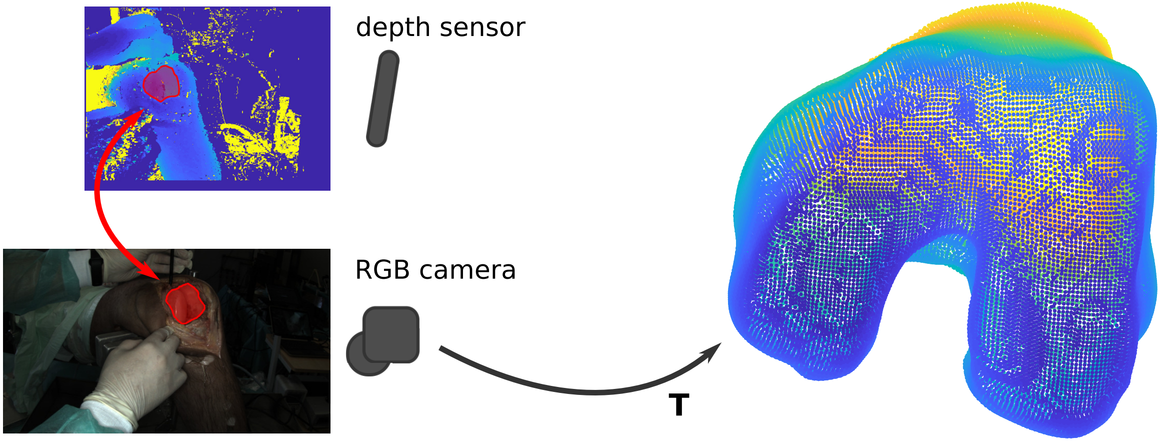


Figure 4.1 – Diagram of the method proposed for markerless surgical navigation. Bone segmentation is performed using frames of the RGB camera. The segmentation is used to get the region of interest in the point cloud from the depth sensor. The point cloud is then registered to the anatomical model to establish the relative pose.

to the bones are bulky, interfering with the normal surgical flow.

This article describes and provides a proof of concept for the first contactless video-based system for computer-aided TKA that does not require any special markers to be attached to the body. A navigation sensor, integrating a consumer RGB camera and a depth camera, is used to register an anatomical model of the patient, obtained with a preoperative computed tomography (CT) scan or magnetic resonance imaging (MRI), such that the bone resections for the implant positioning can be guided according to a preoperative plan. This work introduces the video-based computer-aided TKA system, and describes the two main modules of the software pipeline: (1) bone surface segmentation from RGB images using a deep learning technique, and (2) registration of a preoperative CT/MRI model with a noisy point cloud for computing the pose of the navigation sensor. Augmented reality (AR) techniques for supporting the surgeon in following a preoperative plan are then used. Experimental results in real ex-vivo data are presented.

4.1 Video-based Computer-Aided Surgery

This section overviews the proposed concept for computer-aided TKA that uses a navigation sensor to perform 3D pose estimation during the open surgery. By using additional depth sensing capabilities, we avoid the use of visual markers as in [42] (see figure 4.1).

Given a 3D model of the patient's knee, which was acquired using CT or MRI, the surgeon prepares a preoperative plan for optimizing the implanting positioning, defining the resection plane parameters that will guide the proposed computer-aided TKA system. During the surgery, and at each time instant, RGB and depth data are captured from the navigation sensor and used for computing a local 3D point cloud of the knee joint. The navigation sensor is handheld and is composed of a camera and a depth sensor that are at all times fixed together in a rigid manner. The point cloud is then used to register the 3D preoperative model, enabling the estimation of the pose of the navigation sensor with respect to the model. In this way, the location of the resection planes with respect to the navigation sensor are known, enabling to provide valuable guidance information to the surgeon.

4.1.1 Segmentation of Bone Surface

Image segmentation is a widely investigated topic in medical imaging and computer vision. As in many medical imaging application, in TKA it is difficult to obtain large amounts of accurately labeled data. The main difficulty is that crowd-sourcing frameworks cannot be used, because the labeling of medical data requires expert knowledge and special confidentiality aspects.

We will explore a deep learning approach for image segmentation called U-Net, an encoder-decoder neural network architecture with skip-connections between the encoder and the decoder sections of the network [48]. This network was shown to achieve good results with a relative small number of annotated images for training and for a wide variety of objects and scenarios [48, 27, 28]. The rationale behind the skip-connections is that when doing a traditional encoder-decoder approach some fine-grained details are lost in the encoder, and as the signal is upscaled through the decoder it cannot describe the object in the input image with high resolution. These special connections aim to solve this problem.

The base neural network used was as provided by the authors in [27]. The data processing, the optimization process, along with some other modification, were implemented around the base U-net by Iglovikov and Svets [27]. For the encoder part of the U-Net, the weights where initialized to the weights of the VGG-11 network taking advantage of large datasets of generic training data, similarly to [27]. However, since our training dataset is small, we decided to freeze the encoder weights, while the rest of the network is optimized. This led to better results and faster training. Dropout was implemented for regularization. The hyperparameter space for the optimization is defined by the learning rate, dropout ratio, and the number of filters in the convolutional layers of the decoder. The number of filters in the decoder decreases in a similar way to the encoder growth, but as multiple of this hyperparameter). As for the encoder, the number of filters is fixed, because the weights being used are the same as the weights of the VGG-11 network. A mini-batch of 10 was combined with the Adam optimizer for training the network.

The dataset used in this work contains several video sequences of 2 different femurs (3 sequences for the first femur and 1 sequence for the second femur). The datasets have a wide variety of relative poses and occlusion events. In some of the sequences, a marker was attached to the bone before data acquisition. This base marker was tracked through the sequences so that it could be used to aid in the generation of ground truth data for segmentation and to compare trajectories for evaluation of pose estimation. Due to the dimensions of the dataset (approximately 10,000 images), manual segmentation of all the images was not possible, and a semi-automated approach was used for the labeling task. This was accomplished by manually segmenting approximately 100 images, and propagating the segmentation to the neighboring frames using the detected marker pose and the 3D femur model. The dataset was split into approximately 9,000 images for training and 1,000 for validation. Note that the validation dataset was taken as a different video sequence and not randomly chosen frames of the same sequences to make the validation dataset as different as possible to the training dataset. Although parts of the dataset also contain depth information, only the RGB data was used for learning to segment bone surfaces. For augmenting the variability in the training dataset, we performed particular transformations to the input images on-the-fly. Among these transformation we included: image rotation, imaging flipping, and shifts to the HSV space of the images. Additionally, the base marker was masked and inpainted over to avoid implicit relationships between the poses of the marker and the femur within the learning framework and improve generalization.

4.1.2 Registration of a Preoperative Model with a Noisy Point Cloud

3D registration consists in aligning two models such that their overlapping areas are maximized. It is a well studied problem in computer vision, with applications ranging from SLAM and tracking to robotics and, more recently, to medicine [41]. Some solutions for the 3D registration problem work by matching features extracted from the models and estimating the rigid transformation using

RANSAC or other robust estimators [49, 65]. Such approaches perform poorly when the point clouds are too smooth and/or noisy because of the difficulty in finding repeatable features. The family of algorithms 4PCS [37, 1, 34] makes use of hypothesize-and-test schemes that randomly select sets of 4 coplanar points in one point cloud and find correspondences in the other for establishing alignment hypotheses. Rencently, Raposo and Barreto [40] proposed an algorithm that is faster than the 4PCS family of methods and resilient to very high levels of outliers. In general terms, the method extracts pairs of points and their normals in one point cloud, finds congruent pairs of points in the other and afterwards establishes alignment hypotheses which are tested in a RANSAC scheme. The selected solution is refined using a standard ICP [5] approach.

Closely related to our work in terms of application is [41], which also employs 3D registration in the context of orthopedic surgery for aligning a preoperative model of the targeted bone with the patient's anatomy. However, there are two important differences: (1) while [41] includes an explicit surgical step where the surgeon touches bone surface with a tracked probe for reconstructing 3D points, our method uses a depth sensor and an automatic segmentation process for reconstructing only the area corresponding to the targeted anatomy; (2) [41] requires fiducial markers attached to the patient's body for estimating the camera pose. On the other hand, our approach accomplishes camera pose estimation by registering the segmented point cloud with the preoperative model at each frame. The registration algorithm proposed in [41] is fast, accurate and robust to outliers. However, it is not suitable for our case because it only solves the curve-vs-surface alignment problem and we require surface-vs-surface registration. As mentioned in the previous paragraph, this task is efficiently solved in [40].

Since it is reported that this method is able to handle outliers, we attempted to register the complete point cloud obtained from the depth sensor with the preoperative model. However, due to the significant levels of noise and very high percentages of outliers, the results were not satisfactory, evincing the need for a proper segmentation of the bone surface. The registration parameters were tuned for accommodating the noise in the data and qualitative and quantitative results on the registration accuracy are provided in the next section. Furthermore, some frames contain too many outliers and missing information, whether due to the sensor being too close to the knee (out of range), to specularities and/or total occlusion. Therefore a registration was deemed successful only when 80% of inliers were considered for its computation.

4.2 Experimental Validation

The camera/sensor setup used was composed of a 1080p video camera and a two-camera IR depth sensor with 480p resolution at approximately 10 frames per second. The two components were fixed together and calibrated.
To evaluate the proposed method both quantitatively and qualitatively, we have three datasets: the training and validation datasets used in the machine learning framework, and an additional dataset where no markers where inserted in the bone.
The proposed architecture for femur segmentation obtained an intersection over union (IoU) metric of median 0.853 and a Dice coefficient of median 0.921. The hyperparameters used were: learning rate - 0.0001, epochs - 10, number of filters for the decoder - multiples of 2. Please refer to figures 4.2 and 4.3 for additional results regarding femur segmentation.
To evaluate the registration, we compute the trajectory obtained in the validation dataset and compare it to the trajectory obtained by tracking the marker inserted in the femur. The two trajectories were aligned using a rigid transformation and the results are show in figure 4.4(a). Figure 4.4(b) shows the angular magnitude of the residual rotation between the ground truth and the estimated rotations, in degrees, and the norm of the difference between the ground truth and the obtained translation vectors; where the ground truth was taken as the median pose of all successful registrations, thus giving a measure of robustness of the registration procedure. The obtained median rotation error (eR) is 3.17 degrees and the median translation error (eT) is 6.18 millimeters. In figure 4.5, our contactless registration is used to superimpose the bone model and show the preoperative plan in an AR view.
To further test the generalization power of our method, we performed a new hyperparameter search, this time using only one femur for training (3 video sequences of the first femur) and one for testing (1 video sequence of the second femur). This test aims to show that generalization to other scenes is possible. However, using only one femur for training is not ideal, and for a fully working framework further data is needed. Figure 4.6 shows the results.

4.3 Discussion

The article proposed a new approach for navigation in TKA that avoids the need of attaching fiducials to the anatomy, which is a major problem in current nav-

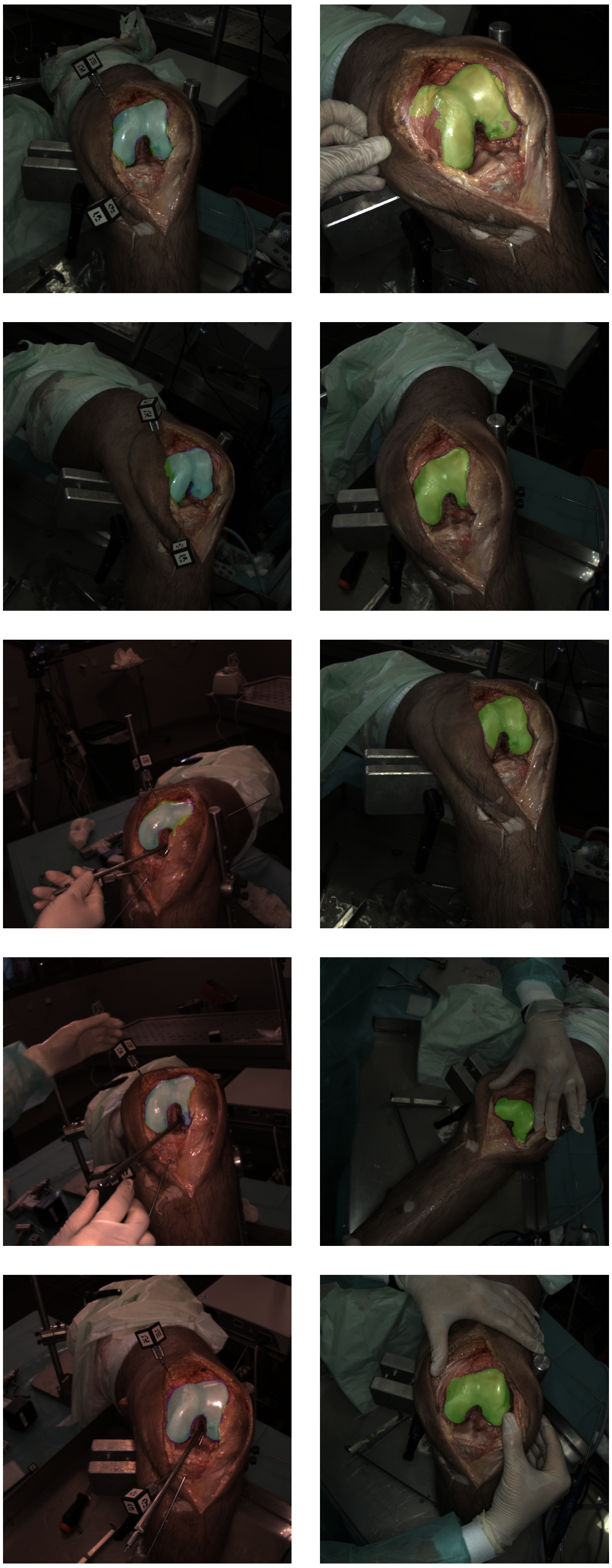
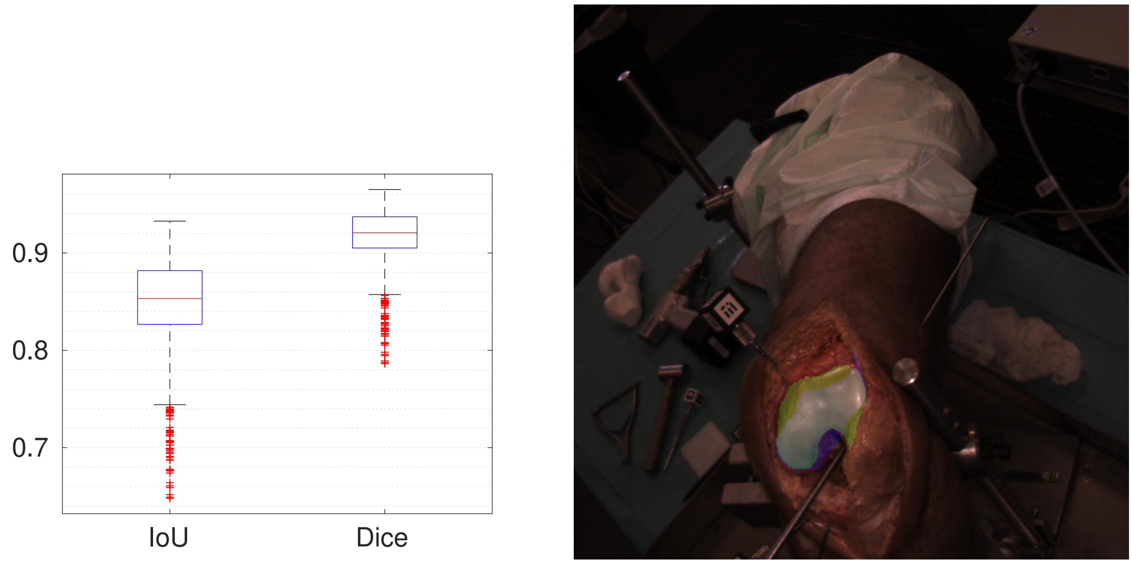


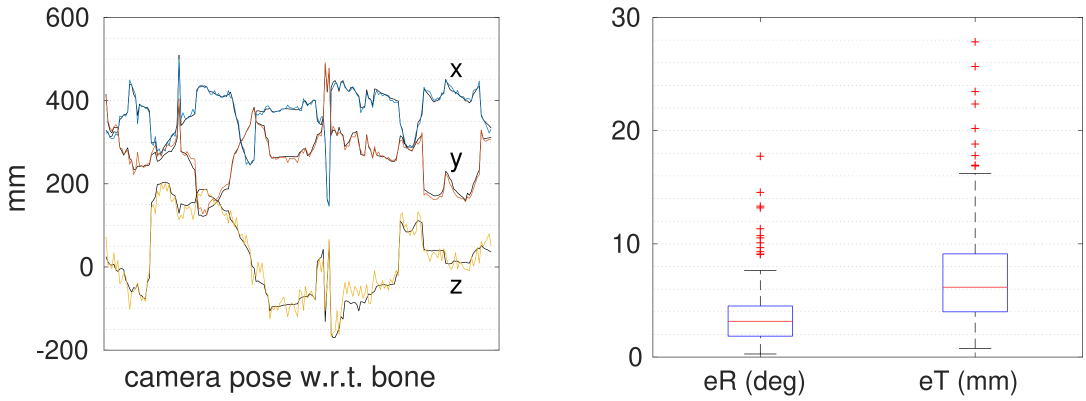
Figure 4.2 – Femur segmentation results on the validation dataset (left column) and on the dataset without markers (right column), which does not have ground truth available. Green: prediction; blue: ground truth; cyan: both.



(a)

(b)

Figure 4.3 – Segmentation metrics for the femur segmentation in the validation dataset: (a) metric distribution; (b) frame with worst IoU metric.



(a)

(b)

Figure 4.4 – Results for the registration between the preoperative model and the segmented point cloud in the validation dataset: (a) comparison between the x, y, and z components of the trajectories of the proposed registration (colors) and marker-based tracking (black); (b) distribution of the rotation error (eR) in degrees and the translation error (eT) in millimeters.

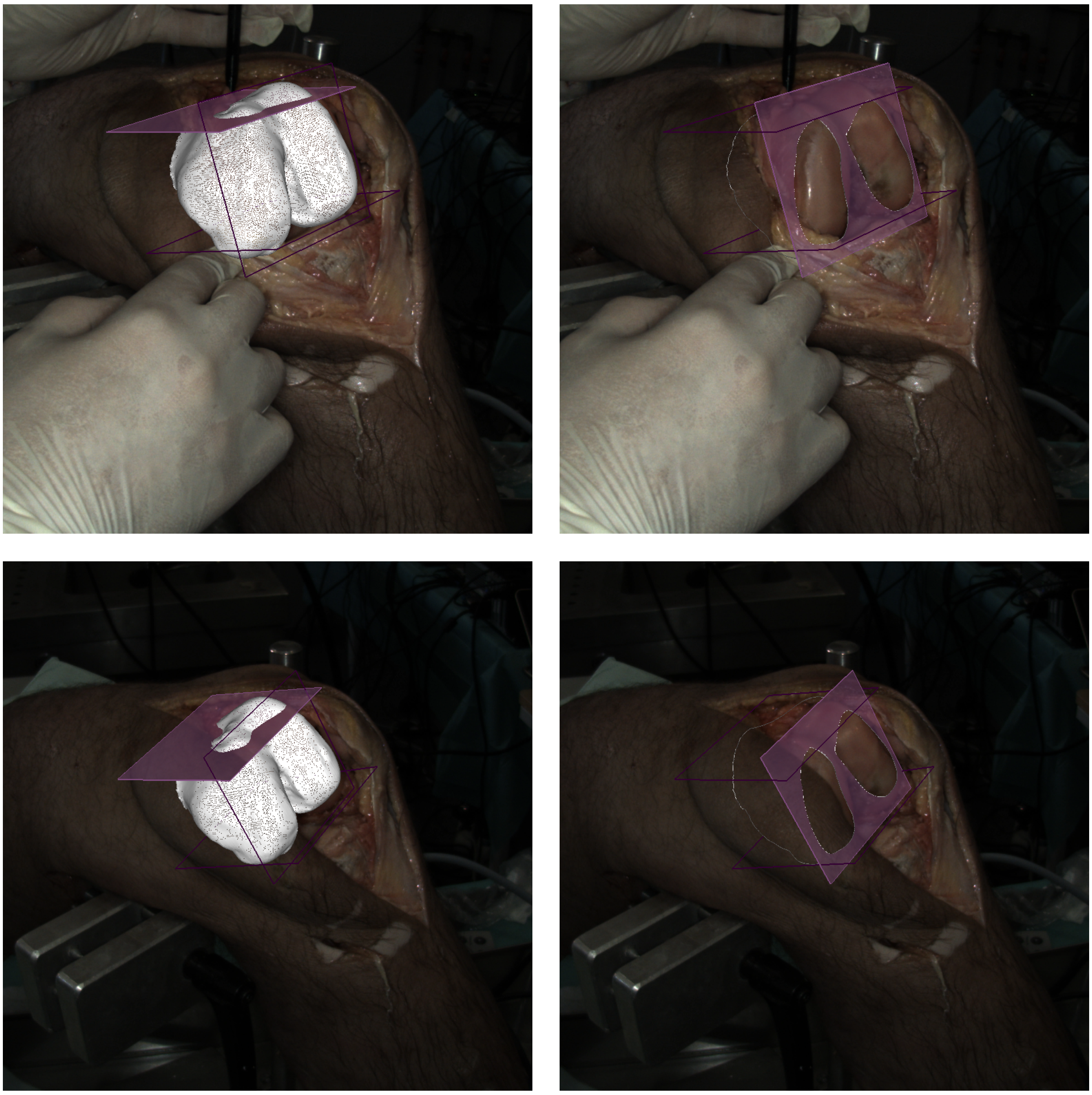


Figure 4.5 – Augmented reality for surgical navigation using our method. Each row corresponds to a different frame showing the markerless and contactless femur registration for AR-guided surgery using preoperative planned cuts.

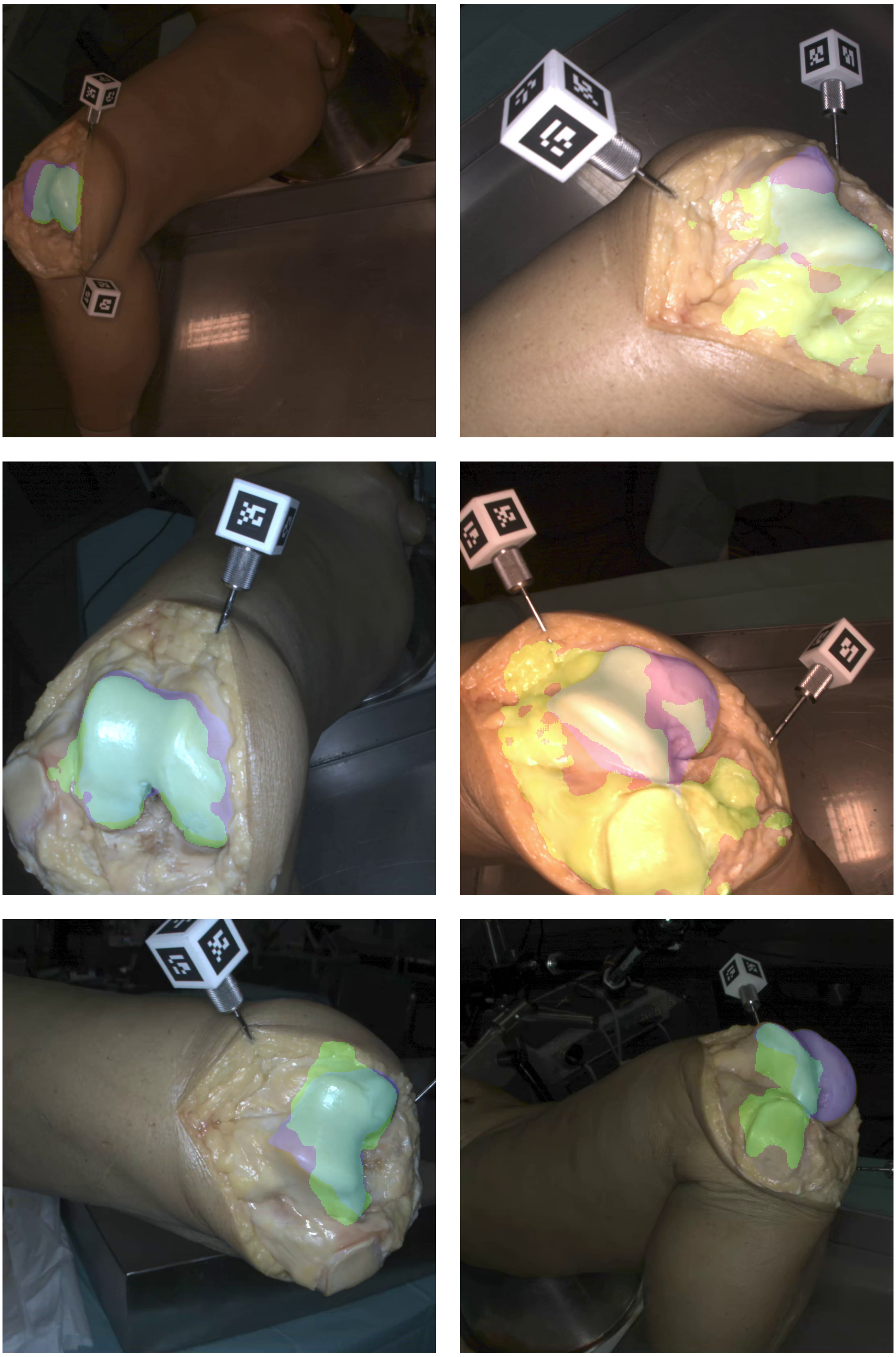


Figure 4.6 – Femur segmentation results using video sequences of only one femur for training: good results (left column) and bad results (right column). Green: prediction; blue: ground truth; cyan: both.

igation systems and cause surgeons to avoid these techniques. Moreover, the proposed approach uses off-the-shelf hardware and does not require any user input.

The segmentation worked under various conditions and surpassed expectations in differentiating between the bone and the adjacent tissue with similar color and texture, even though only approximately 100 images were manually segmented. The scarcity of the data required for performing machine learning tasks, means that fair evaluation of the segmentation algorithm is difficult. Further testing segmentation with additional ex-vivo knees may be necessary to confirm the generalization power shown here.

Regarding the full registration process, the work aimed to be a proof of concept that demonstrated that it is feasible to robustly track the anatomy without the need of attaching fiducial markers. The results are encouraging but there is still work to do to accomplish a final system that can be used in everyday clinical practice:

- Translation errors of 6.18 millimeters and rotation errors of 3.17 degrees, while satisfactory for a proof of concept, are above the requirements for surgical navigation. The obtained errors can lead to critical misalignment of the planned cuts and drills. Future work will address this problem and focus on fine tuning the registration algorithm to work under such extreme depth outlier conditions and possibly use multiple frames to perform the pose estimation. Another promising line of research is to eliminate the need to work with the depth sensor and perform pose estimation with machine learning as well. This is enticing since in our setup the depth map is the main source of imprecision. Another line of research with potential would be to use an end-to-end machine learning approach.

- Occlusion is currently a problem for the segmentation. However, this happens because only 100 manual segmentations were performed. Further manual segmentations can be performed for better resilience to occlusion. Another possible approach is to track the surgical tools and remove them from the segmentation maps to generate the dataset.

- Future work must comprise extension to other anatomies. So far we have worked only with the femur. Extension to other procedures where the anatomy is not so clearly exposed (*e.g.*, hip arthroplasty) may not be as straightforward. Nevertheless, evaluation of accuracy in such cases may be interesting. Extension for the tibia, as required for full TKA navigation, should be feasible but must be validated as well.

Although additional testing is required for a full navigation system to be used in the OR, this work opens the possibility for a contactless registration to be used to guide the surgeon. A possible path for the application of our work is to use the contactless registration to guide the drilling of the holes for the cut guide and only then guide the distal cut. In this way, removing the necessity of the registration the bone after the cuts, which is not contemplated by the present work. However, this approach would require the holes to be drilled before the first cut and further testing is mandatory.

Bibliography

[1] Aiger, D., Mitra, N. J., and Cohen-Or, D. 4-points congruent sets for robust pairwise surface registration. In *ACM SIGGRAPH 2008 papers on - SIGGRAPH '08* (New York, New York, USA, 2008), ACM Press, p. 1. 59

[2] Atcheson, B., Heide, F., and Heidrich, W. CALTag: High Precision Fiducial Markers for Camera Calibration. *Vision, Modeling, and Visualization* (2010), 41–48. 5, 14

[3] Badano, A., Revie, C., Casertano, A., Cheng, W. C., Green, P., Kimpe, T., Krupinski, E., Sisson, C., Skr??vseth, S., Treanor, D., Boynton, P., Clunie, D., Flynn, M. J., Heki, T., Hewitt, S., Homma, H., Masia, A., Matsui, T., Nagy, B., Nishibori, M., Penczek, J., Schopf, T., Yagi, Y., and Yokoi, H. Consistency and Standardization of Color in Medical Imaging: a Consensus Report. *Journal of Digital Imaging 28*, 1 (2014), 41–52. 35

[4] Bala, R., Klassen, R. V., and Braun, K. M. Efficient and simple methods for display tone-response characterization. *Journal of the Society for Information Display 15*, 11 (2007), 947–957. 37

[5] Besl, P., and McKay, N. D. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence 14*, 2 (feb 1992), 239–256. 59

[6] Bhasker, E., Juang, R., and Majumder, A. Registration techniques for using imperfect and partially calibrated devices in planar multi-projector displays. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007), 1368–1375. 37

[7] Bhasker, E. S., Sinha, P., and Majumder, A. Asynchronous distributed calibration for scalable and reconfigurable multi-projector displays. *IEEE Transactions on Visualization and Computer Graphics 12*, 5 (2006), 1101–1108. 37

[8] BOURNE, R. B., CHESWORTH, B. M., DAVIS, A. M., MAHOMED, N. N., AND CHARRON, K. D. J. Patient Satisfaction after Total Knee Arthroplasty: Who is Satisfied and Who is Not? *Clinical Orthopaedics and Related Research® 468*, 1 (jan 2010), 57–63. 55

[9] BRAINARD, D. H., PELLI, D. G., AND ROBSON, T. Display characterization. In *Encyclopedia of Imaging Science and Technology*. John Wiley & Sons, Inc., Hoboken, NJ, USA, jan 2002. 36, 40

[10] BROWN, M., MAJUMDER, A., AND YANG, R. Camera-based calibration techniques for seamless multiprojector displays. *IEEE Transactions on Visualization and Computer Graphics 11*, 2 (2005), 193–206. 36

[11] CHAKRABARTI, A., SCHARSTEIN, D., AND ZICKLER, T. An Empirical Camera Model for Internet Color Vision. *Procedings of the British Machine Vision Conference 2009* (2009), 51.1–51.11. 3, 10

[12] CIE. Chromaticity difference specification for light sources, 2014. (accessed 01 October 2019). 43

[13] COffiN, D. Decoding raw digital photos in Linux, 2015. (accessed 01 October 2019). 40

[14] COLLINS, T., AND BARTOLI, A. 3D Reconstruction in Laparoscopy with Close-Range Photometric Stereo. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2012*, N. Ayache, H. Delingette, P. Golland, and K. Mori, Eds., vol. 15 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, jan 2012, pp. 634–642. 15

[15] DEBEVEC, P. E., AND MALIK, J. Recovering high dynamic range radiance maps from photographs. In *ACM SIGGRAPH 1997* (New York, New York, USA, 1997), ACM Press, p. 1. 6, 7

[16] DÍAZ, M., AND STURM, P. Radiometric calibration using photo collections. In *2011 IEEE International Conference on Computational Photography (ICCP)* (apr 2011), IEEE, pp. 1–8. 6, 7

[17] FINLAYSON, G. D., MOHAMMADZADEH DARRODI, M., AND MACKIEWICZ, M. The alternating least squares technique for nonuniform intensity color correction. *Color Research and Application 40*, 3 (jun 2015), 232–242. 6, 8

[18] FORSYTH, D. A. A novel algorithm for color constancy. *International Journal of Computer Vision 5*, 1 (aug 1990), 5–35. 8, 9

[19] Frigo, O., Sabater, N., Delon, J., and Hellier, P. Motion Driven Tonal Stabilization. *IEEE Transactions on Image Processing 25*, 11 (2016), 5455–5468. 8

[20] Funt, B., and Bastani, P. Irradiance-independent camera color calibration. *Color Research and Application 39*, 6 (dec 2014), 540–548. 6, 8

[21] Goldman, D. B. Vignette and exposure calibration and compensation. *IEEE transactions on pattern analysis and machine intelligence 32*, 12 (dec 2010), 2276–88. 8

[22] Gonçalves, N., Roxo, D., Barreto, J. P., Rodrigues, P., Gonçalves, N., and Barreto, J. P. Perspective shape from shading for wide-FOV near-lighting endoscopes. *Neurocomputing 150* (oct 2015), 21–30. 2, 3

[23] Grossberg, M. D., and Nayar, S. K. Modeling the space of camera response functions. *IEEE transactions on pattern analysis and machine intelligence 26*, 10 (oct 2004), 1272–82. 21

[24] Grundmann, M., McClanahan, C., Kang, S. B., and Essa, I. Post-processing approach for radiometric self-calibration of video. *IEEE International Conference on Computational Photography (ICCP)* (2013), 1–9. 10

[25] Haneishi, H., Shiobara, T., and Miyake, Y. Color correction for colorimetric color reproduction in an electronic endoscope. *Optics Communications 114*, 1-2 (1995), 57–63. 6, 8

[26] Hardeberg, J. Y., Seime, L., and Skogstad, T. Colorimetric characterization of projection displays using a digital colorimetric camera. In *Procedings of SPIE on Projection Displays IX* (mar 2003), M. H. Wu, Ed., vol. 5002, pp. 51–61. 36

[27] Iglovikov, V., and Shvets, A. TernausNet: U-Net with VGG11 Encoder Pre-Trained on ImageNet for Image Segmentation. 57

[28] Isensee, F., Petersen, J., Kohl, S. A. A., Jäger, P. F., and Maier-Hein, K. H. nnU-Net: Breaking the Spell on Successful Medical Image Segmentation. 1–8. 57

[29] Jung, J. Y., Kim, S. W., Park, S., Choi, B. D., and Ko, S. J. Camera-based color calibration method for multiple flat-panel displays using smartphone. *Journal of Display Technology 12*, 12 (2016), 1777–1784. 37, 41, 48, 53

[30] **Kim, S. J., and Pollefeys, M.** Robust radiometric calibration and vignetting correction. *IEEE transactions on pattern analysis and machine intelligence 30*, 4 (apr 2008), 562–76. 6, 7

[31] **Kurtz, S., Ong, K., Lau, E., Mowat, F., and Halpern, M.** Projections of Primary and Revision Hip and Knee Arthroplasty in the United States from 2005 to 2030. *The Journal of Bone & Joint Surgery 89*, 4 (apr 2007), 780–785. 55

[32] **Lin, S., and Yamazaki, S.** Radiometric calibration from a single image. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.* (2004), vol. 2, IEEE, pp. 938–945. 6, 7

[33] **Matsushita, Y., and Lin, S.** Radiometric Calibration from Noise Distributions. In *2007 IEEE Conference on Computer Vision and Pattern Recognition* (jun 2007), Ieee, pp. 1–8. 6, 7

[34] **Mellado, N., Aiger, D., and Mitra, N. J.** Super 4PCS Fast Global Pointcloud Registration via Smart Indexing. *Computer Graphics Forum 33*, 5 (aug 2014), 205–215. 59

[35] **Melo, R., Barreto, J. P., and Falcão, G.** A new solution for camera calibration and real-time image distortion correction in medical endoscopy-initial technical evaluation. *IEEE transactions on bio-medical engineering 59*, 3 (mar 2012), 634–44. 5, 14, 15

[36] **Mitsunaga, T., and Nayar, S.** Radiometric self calibration. In *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)* (1999), IEEE Comput. Soc, pp. 374–380. 6, 7

[37] **Mohamad, M., Ahmed, M. T., Rappaport, D., and Greenspan, M.** Super Generalized 4PCS for 3D Registration. In *2015 International Conference on 3D Vision* (oct 2015), IEEE, pp. 598–606. 59

[38] **Ng, T.-T., Chang, S.-F., and Tsui, M.-P.** Using Geometry Invariants for Camera Response Function Estimation. In *2007 IEEE Conference on Computer Vision and Pattern Recognition* (Minneapolis, MN, jun 2007), IEEE, pp. 1–8. 6, 7

[39] **Post, M., Fieguth, P., Naiel, M. A., Azimifar, Z., and Lamm, M.** FRESCO: Fast radiometric egocentric screen compensation. In *2019 IEEE/CVF Confer-*

ence on Computer Vision and Pattern Recognition Workshops (CVPRW) (jun 2019), IEEE, pp. 1899–1906. 37

[40] **Raposo, C., and Barreto, J. P.** Using 2 point+normal sets for fast registration of point clouds with small overlap. In *2017 IEEE International Conference on Robotics and Automation (ICRA)* (may 2017), IEEE, pp. 5652–5658. 59

[41] **Raposo, C., and Barreto, J. P.** 3D Registration of Curves and Surfaces Using Local Differential Information. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2018). 58, 59

[42] **Raposo, C., Sousa, C., Ribeiro, L., Melo, R., Barreto, J. P., Oliveira, J., Marques, P., and Fonseca, F.** Video-Based Computer Aided Arthroscopy for Patient Specific Reconstruction of the Anterior Cruciate Ligament. In *MICCAI* (2018), pp. 125–133. 56

[43] **Rodrigues, P., Antunes, M., Raposo, C., Marques, P., Fonseca, F., and Barreto, J.** Deep segmentation leverages geometric pose estimation in computer-aided total knee arthroplasty. *Healthcare Technology Letters* (oct 2019), 1–5. 2

[44] **Rodrigues, P., and Barreto, J. P.** Methods and Apparatus for Estimating Camera Response Function and Vignetting Under Non-Uniform Illumination, apr 2015. 2

[45] **Rodrigues, P., and Barreto, J. P.** Single-image estimation of the camera response function in near-lighting. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (jun 2015), IEEE, pp. 1028–1036. 2, 4, 5, 6, 14, 15, 19, 22

[46] **Rodrigues, P., Barreto, J. P., and Antunes, M.** Color reproduction and characterization of display monitors using a camera (submitted). *Displays*. 2

[47] **Rodrigues, P. M., Barreto, J. P., and Antunes, M.** Photometric camera characterization from a single image with invariance to light intensity and vignetting. *Computer Vision and Image Understanding 192* (mar 2020), 102887. 2, 6

[48] **Ronneberger, O., Fischer, P., and Brox, T.** U-Net: Convolutional Networks for Biomedical Image Segmentation. 2015, pp. 234–241. 57

[49] RUSU, R. B., BLODOW, N., AND BEETZ, M. Fast Point Feature Histograms (FPFH) for 3D registration. In *2009 IEEE International Conference on Robotics and Automation* (may 2009), IEEE, pp. 3212–3217. 59

[50] SCOTT, W. N. *Insall&Scott Surgery of the Knee*. Elsevier, 2017. 55

[51] SEIME, L., AND HARDEBERG, J. Y. Colorimetric characterization of LCD and DLP projection displays. *Journal of the Society for Information Display 11*, 2 (2003), 349–358. 36

[52] SEON JOO KIM, HAI TING LIN, ZHENG LU, SÜSSTRUNK, S., LIN, S., AND BROWN, M. S. A New In-Camera Imaging Model for Color Computer Vision and Its Application. *IEEE Transactions on Pattern Analysis and Machine Intelligence 34*, 12 (dec 2012), 2289–2302. 3, 4, 6, 7, 10, 12

[53] SHARMA, G., WU, W., AND DALAL, E. N. The CIEDE2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations. *Color Research & Application 30*, 1 (feb 2005), 21–30. 25

[54] THOMAS, J. B. *Colorimetric Characterization of Displays and Multi-display Systems*. PhD thesis, Université de Bourgogne, 2009. 37, 40, 41

[55] VAN DER LIST, J. P., CHAWLA, H., JOSKOWICZ, L., AND PEARLE, A. D. Current state of computer navigation and robotics in unicompartmental and total knee arthroplasty: a systematic review with meta-analysis. *Knee Surgery, Sports Traumatology, Arthroscopy 24*, 11 (nov 2016), 3482–3495. 55

[56] VAZQUEZ-CORRAL, J., AND BERTALMÍO, M. Color stabilization along time and across shots of the same scene, for one or several cameras of unknown specifications. *IEEE Transactions on Image Processing 23*, 10 (2014), 4564–4575. 8

[57] VAZQUEZ-CORRAL, J., AND BERTALMIO, M. Simultaneous blind gamma estimation. *IEEE Signal Processing Letters 22*, 9 (2015), 1316–1320. 8

[58] VAZQUEZ-CORRAL, J., AND BERTALMIO, M. Log-encoding estimation for color stabilization of cinematic footage. In *2016 IEEE International Conference on Image Processing (ICIP)* (sep 2016), IEEE, pp. 3349–3353. 8

[59] VAZQUEZ-CORRAL, J., CONNAH, D., AND BERTALMÍO, M. Perceptual Color Characterization of Cameras. *Sensors 14*, 12 (2014), 23205–23229. 10

[60] Visentini-Scarzanella, M., Stoyanov, D., and Yang, G.-Z. Metric depth recovery from monocular images using Shape-from-Shading and specularities. In *2012 19th IEEE International Conference on Image Processing* (sep 2012), IEEE, pp. 25–28. 30

[61] Wilburn, B., Xu, H., and Matsushita, Y. Radiometric calibration using temporal irradiance mixtures. In *2008 IEEE Conference on Computer Vision and Pattern Recognition* (jun 2008), Ieee, pp. 1–7. 6, 7

[62] Wu, C., Jaramaz, B., and Narasimhan, S. G. A full geometric and photometric calibration method for oblique-viewing endoscopes. *Computer Aided Surgery* (2010). 5

[63] Wu, C., Narasimhan, S. G., and Jaramaz, B. A Multi-Image Shape-from-Shading Framework for Near-Lighting Perspective Endoscopes. *International Journal of Computer Vision 86*, 2-3 (feb 2010), 211–228. 5, 6, 7, 22, 25, 27, 33

[64] Wu, W., and Allebach, J. P. Imaging Colorimetry Using a Digital Camera. *Journal of Imaging Science and Technology* 44, 4 (2000), 267–279. 9

[65] Zhou, Q.-Y., Park, J., and Koltun, V. Fast Global Registration. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9906 LNCS. 2016, pp. 766–782. 59