Improved color acquisition and mapping on 3D models via flash-based photography

M. Dellepiane, M. Callieri, M. Corsini, P. Cignoni and R. Scopigno
Visual Computing Lab, ISTI-CNR Pisa Italy

Flash light of digital cameras is a very useful way to picture scenes with low quality illumination. Nevertheless, especially low-end cameras integrated flash lights are considered as not reliable for high quality images, due to known artifacts (sharp shadows, highlights, uneven lighting) generated in images. Moreover, a mathematical model of this kind of light is difficult to create. In this paper we present a color correction space which, given some information about the geometry of the pictured scene, is able to provide a space-dependent color correction for each pixel of the image. The correction space can be calculated once in a lifetime using a quite fast acquisition procedure; after 3D spatial calibration, the obtained color correction function can be applied to every image where flash is the dominant light source. We developed this approach to produce better color samples in the application framework of color mapping on 3D scanned models. The correction space proposed presents several advantages: it is independent from the kind of light used (provided that it is bound to the camera), it gives the possibly to correct some artifacts (for example color deviation) introduced by flash light, and it has a wide range of possible applications, from image enhancement to material color estimation. Moreover, once that the inverse photo-to-geometry transformation is known, it allows the easy estimation of the flash light position and permits to identify and remove other annoying artifacts, like highlights and shadows. The resulting approach allows to gather in an easy manner a better and more consistent color information and to produce higher quality 3D models.

Categories and Subject Descriptors: I.4.1 [**Digitization and Image Capture**]: Scanning; I.3.7 [**Three-Dimensional Graphics and Realism**]: Color, shading, shadowing, and texture; I.3.5 [**Computational Geometry and Object Modeling**]: Geometric algorithms, languages, and systems

General Terms: Algorithms

Additional Key Words and Phrases: Geometry acquisition and processing, color mapping, out-of-core data management

1. INTRODUCTION

3D scanning has become a widely used technology for the acquisition of highly accurate geometric data from real objects. The initial issues related to the management of the very dense sampling of geometric data have been mostly overcome in recent years, thanks to several new approaches to encode, process and render

Author's address: Visual Computing Lab, ISTI - CNR, v. Moruzzi 1, 56100 Pisa, Italy
Email: name.surname@isti.cnr.it

the sampled data. But shape acquisition is only one side of the problem. The aim could be to add high quality appearance information to the geometry data. Many current applications require also an accurate sampling of surface reflection properties to perform a number of useful operations (high-quality rendering, relighting, color projection on copies [Raskar et al. 2001]). Seminal works have proposed approaches to sample Bidirectional Radiance Distribution Functions (BRDF) [Lensch et al. 2003; Debevec et al. 2000] by means of sophisticated controlled lighting environments.
Unfortunately, there are applications where the objects of interest cannot be transferred to a reflection acquisition lab. Cultural Heritage is a fitting example: we need to sample many artifacts which usually cannot be moved from their location (e.g. a museum room); moreover, budget and sustainability considerations impose the use of low cost and easy-to-use procedures and technologies.

An easy solution to acquire a lot of information about the appearance of the target object is to use digital photographs. In a similar fashion to scanning campaign, an object can be entirely depicted in a very short time: moreover, due to the high resolution provided by digital cameras, a few tens of images can be enough to cover the whole surface of a complex objects with a high sampling density.
However, the projection of a set of images on a 3D model need to solve several sub-problems, such as the image-to-geometry registration, color mapping and integration on the 3D mesh, and finally visualization of very dense shape+color models. Usually, the quality of the final colored 3D model is strongly related to the quality of the starting photo set. Most of the photographic artifacts (e.g. color intensity discontinuities, specular highlights and shadows) projected on the 3D model are generated by the specific illumination of the scene, which usually differs from one photo to the other. These kinds of artifacts can be removed by knowing exactly the lighting environment at the time of the shot. Unfortunately, it is usually quite hard to recover the position and contribution of all the light sources in the scene, without introducing specific techniques that uses probes [Debevec 1998; Corsini et al. 2008] and complicating the photo acquisition and processing phases. In effect, most of the controlled light setup solutions are difficult to apply in practical applications and in unconstrained conditions.

In this paper, we propose a method to automatically correct illumination artifacts from images produced by using a very simple controlled light setup: the *camera flash light*. In particular, a procedure which is able to correct the color values of the acquired images is combined with a method able to remove highlights and shadows. The main contributions of this work are:

(1) A simple procedure, needed only once in a camera lifetime, to estimate:
    —the flash position with respect to the camera lenses;
    —a color correction space, where a correction matrix is associated to each point in the camera field of view.
    Assuming that a previous image-to-geometry transformation has been computed for each photograph, to reproject it in the 3D scene, this approach permits to correct the color of each pixel according to the position of the corresponding point in space, removing the deviation introduced by the flash light.
(2) An automatic method which allows to improve the integration of the (usually

redundant) color samples, by not taking into account in the integration phase the specular highlights and the shadows contained in the acquired images.

2. RELATED WORK

The work proposed in this paper is related to several topics in Computer Graphics and Computer Vision research: controlled light environments, light modeling, material properties acquisition, computational photography.
In the context of this Section, we will focus on three of the most relevant subjects: color constancy and white balance, digital photography (with particular aim to illumination artifacts removal and use of flash light) and color information acquisition and mapping.

Color constancy and white balance. White balance is a key issue in the context of the *color constancy problem*, that studies the constancy of perceived colors of surfaces under changes in the intensity and spectral composition of the illumination. Several works in this field rely on the assumption that a single illuminant is present: the enhancement of photos can be based on geometric models of color spaces [Finlayson and Hordley 2000], statistical analysis of lights and colors [Finlayson et al. 2001] or natural images [Gijsenij and Gevers 2007], study of the edges of the image [van de Weijer and Gevers 2005].
Another group of papers deals with mixed lighting conditions. Methods can be semi-automatic [Lischinski et al. 2006] or automatic. Automatic methods usually work well under quite strong assumptions, like hard shadows and black-body radiators lights [Kawakami et al. 2005] or localized gray-world model [Ebner 2004]. A very recent work [Hsu et al. 2008] proposes a white balance technique which renders visually pleasing images by recovering a set of dominant material colors using the technique proposed by [Omer and Werman 2004]. One of the assumptions is that no more than two light types (specified by the user) illuminate the scene. Most of the cited works share some of the main hypotheses of our method. Nevertheless, the knowledge of some information about the geometry of the scene eliminates the need for other restricting assumptions (such as smooth illumination, gray-world theory, need of user interaction).

Artifacts removal and Flash/No-Flash use in Digital Photography. The removal of artifacts from images is an operation which can be valuable for several fields of application, hence it has been widely studied. There are a number of *highlights removal* techniques which have been proposed in the last few years. They can be roughly divided in two subgroups: the ones working on a single image [Wolff 1989; Tan et al. 2003; Ortiz and Torres 2006; Shen et al. 2008], which are mainly based on the analysis of the colors of the image, and the ones using a set of images [Sato and Ikeuchi 1993; Lin et al. 2003], which take advantage of the redundancy of information between images. In general, these methods assume no prior information about the geometry of the scene.
More recently, the use of flash/no-flash pairs to enhance the appearance of photographs has been proposed in several papers. The *continuous flash* [Hoppe and Toyama 2003] has been a seminal work, where flash and no-flash images are com-

bined to create adjustable images. Two almost contemporaneous papers [Eisemann and Durand 2004; Petschnigg et al. 2004] proposed techniques to enhance details and reduce noise in ambient images, by using flash/no-flash pairs. These works, which mainly differ only in the treatment of flash shadows, provide features for detail transfer, color and noise correction, shadows and highlights removal. Results are very interesting, considering the lack of geometry information, but clearly the systems are not completely automatic. The goal of a more recent work [Agrawal et al. 2005] is to enhance flash photography: in addition to the techniques just mentioned, a flash imaging model is proposed, and a gradient projection scheme is used to reduce the visual effects of noise. Moreover, by taking several images at different exposures and flash intensities a HDR image is created and used to enhance the results. Flash/no-flash pairs are used by [Lu et al. 2006] to detect and remove ambient shadows.

Mapping of color information on 3D models. The apparent color value, as sampled in digital photos, is mapped on the digital object surface by registering those photos w.r.t. the 3D model (computing the camera parameters) and then by applying inverse projection, transferring the color from the images to the 3D surface. Despite the simple approach, there are numerous difficulties in selecting the correct color to be applied (when multiple candidates are present among different images), dealing with discontinuities caused by color differences between photos that cover adjacent areas and reducing the illumination-related artifacts (shadows, highlights, peculiar BRDFs).

One of the main issues in the color mapping field is the color storage. However, in the framework of this paper we focus on the problems related to solving image discrepancies and to reduce illumination artifacts.

A first method to decide which color has to be applied to a particular area of the model is to select for each part of the surface an image following a particular criteria that, in most of cases [Callieri et al. 2002; Bannai et al. 2004; Lensch et al. 2000], is the orthogonality between the surface and the view direction. In this way, only the "best" parts of the images are chosen and processed. Artifacts caused by the discordance between overlapping images are then visible on the border between surface areas that receive color from different images. Between those adjacent images there is a common, redundant zone: this border can be used to obtain an adequate corrections in order to prevent sharp discontinuities. This approach was followed by [Callieri et al. 2002], who propagates the correction on the texture space, and by [Bannai et al. 2004], who used the redundancy to perform a matrix-based color correction on the original image. Other approaches, like the one proposed by [Lensch et al. 2000] do not work only on the frontier area, but blend on the 3D surface using the entire shared content to smooth out the discontinuities.

Instead of cutting and pasting parts of the original images, as the previous approach have done, it is possible to assign a weight to each input pixel (this value expresses the "quality" of its contribution), and to select the final color of the surface as the weighted mean of the input data, as in [Pulli et al. 1998]. The weight is usually a combination of various quality metrics. This weight-blend strategy has been introduced, with many variants in terms of number and nature of assembled

metrics, in various papers [Bernardini et al. 2001; Baumberg 2002; Rankov et al. 2005]. In particular, [Callieri et al. 2008] presented a flexible weighting system, that could be extended in order to accommodate additional metrics.

Most of the analyzed methodologies present a common operation: the possibility to *discard* parts of the input images or to selectively *assign a weight* to contributing pixels. Since we will be able to precisely detect image artifacts produced by the flash, this detection would be a nice addition to the aforementioned methods, making them able to reduce the impact of such artifacts by eliminating or reducing the weight of unwanted parts.

3. CORRECTION OF IMAGES UNDER FLASH ILLUMINATION

Flash light is generally a deprecated kind of illumination for non professional photographers for two main reasons: the production of undesirable artifacts and the variability of its effect between photos. These disadvantages can be partially solved using expensive type of flashes and more complex light settings.
Nevertheless, an extremely interesting aspect of the use of flash light is that the source of illumination is constrained to the camera, so that once the image is aligned to the corresponding 3D model, the position of the flash can be automatically found. The main flash artifacts which must be corrected in order to obtain high quality color information are: uneven lighting, color deviation, highlights and sharp shadows. The aim is to be able to automatically correct them once that an image is registered to the 3D model.
In the next subsections, we will describe the operations needed to collect the basic calibration data which will be used to correct the artifacts, and the methods used to perform the correction. One of the main requirements is that the data acquisition operations should be performed only once in a camera lifetime, in order to be able to perform corrections in a very easy way.

3.1 The color correction space

The first aim of our work is to build a spatial color correction function that associates a specific color correction procedure to each point in the camera frustum space. We call this particular data structure *color correction space*. Such an approach allows to override the limitations assumed in most of the color correction approaches [Barnard et al. 2002; Barnard et al. 2002], that is that the illumination is constant, or easily model-able, across the scene. Our main assumptions are: flash light can be considered the dominant illumination in the scene; the light interaction can be described using just sRGB space (we do not account full spectra data); surfaces are non-emitting.
These hypotheses are common among existing techniques which deal with single illumination, and they cover most of the real cases.

3.2 The creation of the color correction space

Typically, the color calibration of digital photographs consists in taking a snapshot of a pre-defined color reference target, such as a Macbeth ColorChecker^{TM} or an AGFA IT8^{TM}, placed near the subject of interest, and estimating the parameters of a transformation that maps the colors of the reference target in the image into its real colors.

A quite simple transformation to model the color correction is a linear affine one, $c' = Ac + B$. Obviously, due to the nonlinear nature of image color formation, this kind of correction is a rough approximation and many other approaches could be used [Barnard et al. 2002; Barnard et al. 2002]. Moreover, the correction is effective for the image parts that are close (and with a similar illumination) to the reference target. On the other hand, in practice, this simple and compact approach works reasonably well in most cases. The linear transformation can be written as a 4×3 matrix:

$$\begin{pmatrix} R' \\ G' \\ B' \end{pmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & c_{34} \end{bmatrix} \begin{pmatrix} R \\ G \\ B \\ 1 \end{pmatrix} \tag{1}$$

where (R, G, B) is the color to correct, (R', G', B') is the corrected color and c_{ij} are the correction parameters. In the following we refer to the matrix formed by the c_{ij} elements as the color correction matrix $\mathcal{C}$.
Roughly speaking, the parameters of $\mathcal{C}$ have the following meaning: (c_{11}, c_{22}, c_{33}) are related to the change in contrast of the color; $(c_{12}, c_{13}, c_{21}, c_{23}, c_{31}, c_{32})$ are related to the color deviation caused by the color of the flash light (if the flash is purely white light, these components tend to zero); (c_{14}, c_{24}, c_{34}) are related to the intensity offset. We use the term contrast in the sense that the multiplication for the coefficients expands the range of values of the channels.
Given the assumptions above, we can finally define our color correction space, which is a space where a color correction matrix is associated to each point in a camera frustum. The correction space will be calculated starting from several sampled points in the camera space. The process of correcting an image will, for each pixel in the image, use the appropriate correction matrix according its corresponding position in the camera space. Due the continuous nature of the correction space, the correction will prove to be reliable even without a precise digital model of the scene, so that an approximate reconstruction such as the ones generated, for example, by stereo matching could be used as well.

The aim of our work was to try to build a procedure which could be used in a general case, without using prototypal or expensive devices. The computation of our color correction space is necessary only once (or very few times) in a camera lifetime. But even with this assumption, it was important to define some as simple and fast as possible acquisition procedures. Hence, we decided to sample the camera space view frustum by taking flash lighted photos of a small color target, calculating the correction matrix in those points and subsequently building the entire space by interpolation. We performed the acquisition with three different models of digital cameras, shown in Figure 1(left). These models are representative of three categories of non-professional cameras: compact, digital SLR (single-lens reflex) and digital SLR with external flash.
As a color target we used a Mini Macbeth ColorChecker; its size (about $3.5'' \times 2.5''$) allows the assumption that the light variation across it is negligible. To sample the view frustum of the camera we sliced it with several planes of acquisition at different distances, moving the Mini Macbeth in different positions for each plane. We divided a distance range between 50 and 220 cm in 7 planes, with 25 positions for

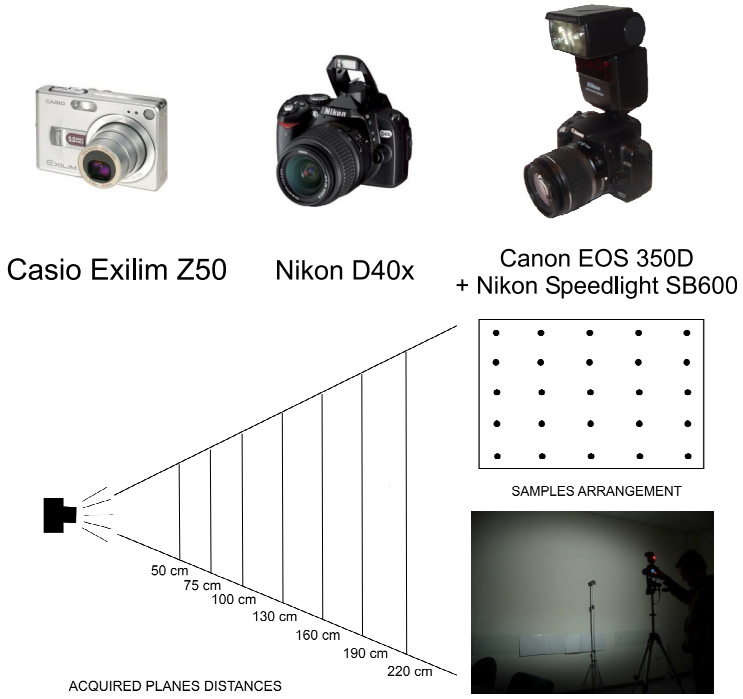

Fig. 1. Top: Digital cameras used for light space sampling. Bottom: the scheme of acquisition for flash space sampling, with a snapshot of the acquisition setup.

each plane, as shown in Figure 1(right). The color target was placed on a tripod, and always faced towards the camera. For each position multiple snapshots were taken, in order to deal with the known variability of flash behavior, and keeping a fixed exposure time and aperture during the entire procedure.
The snapshots were acquired in sRGB RAW format to avoid any other internal processing by the digital camera, except for the Casio compact camera, with which we were forced to use JPEG images. The acquisition procedure of all the needed images for each model took a couple of hours. The processing of acquired data was subdivided in two main phases. In the first one we calibrated all the acquired images, using the color target reference. In the second phase we built the color correction space through parameters interpolation. Figure 2 shows a schematization of the entire data processing.

3.2.1 *Color Calibration..* The color calibration was done using the calibration model previously explained. The parameters of the matrix $\mathcal{C}$ were estimated by solving a linear system with the c_{ij} as unknowns. Our parameter estimation algorithm takes inspiration from the RANSAC approach.
Since four colors could be sufficient to write a system with 12 equations and 12 unknowns, several random combinations of colors are used to find the best solution in terms of quality. The quality of the color correction is evaluated considering the

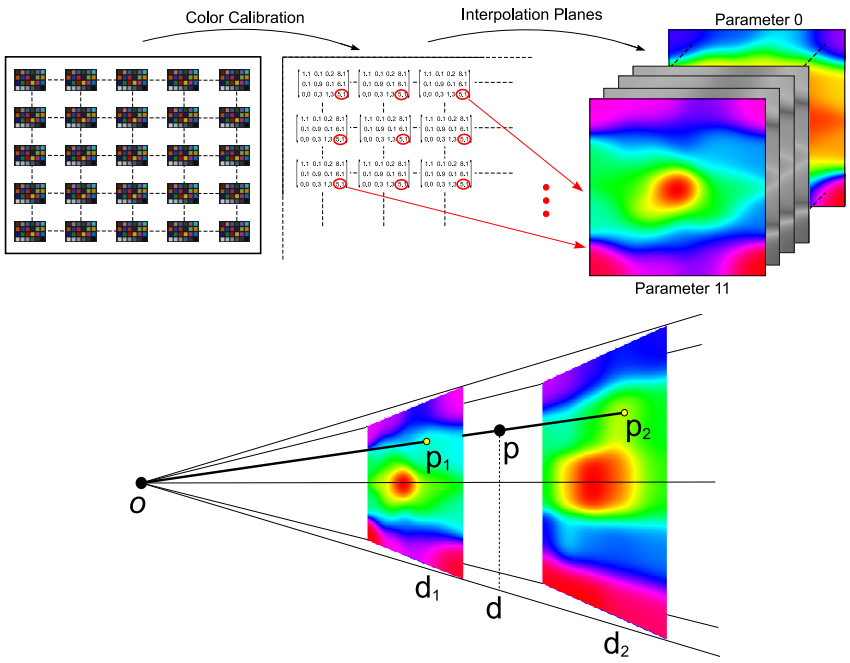

Fig. 2. Data processing. Top: the acquired color targets are calibrated generating a set of color correction matrices, and then interpolated on each plane. Bottom: each correction parameter is interpolated to fill the whole camera space.

CIELab distance [Commission Internationale de l'Eclairage (CIE) 2004] between the real colors and the corrected ones. Hence, after extracting the Mini Macbeth in a semi-automatic way and segmenting each color patch, several permutations of the colors are tried in order to find the best estimation in terms of quality. This robust approach produces better results from a perceptual point of view than a standard least square approach.
After calibration, each image is associated to: the 12 parameters of the matrix $\mathcal{C}$, the position of the Mini Macbeth relative to the camera view and the distance at which the image has been taken. Since several shots of the same position of the target are taken, obtained values are the means of all the obtained estimations. Before this, the data are further processed with a statistical analysis [Grubbs 1969] to remove the outliers from the acquired data.

3.2.2 *Data Interpolation..* Starting from a set of color correction matrices for several points in the space, we could try either to fit a mathematical function or to calculate the intermediate color correction matrix through interpolation. Here, we opt for interpolation, leaving the first approach as an interesting direction of future research. Given a point p in camera space, we calculate the corresponding color correction matrix as the linear interpolation in the squared distance between

the camera center O and the point p in the following way:

$$\mathcal{C}_{ij}(p) = \mathcal{C}_{ij}(p_1) + (d^2 - d_1^2)\frac{\mathcal{C}_{ij}(p_2) - \mathcal{C}_{ij}(p_1)}{d_2^2 - d_1^2} \tag{2}$$

where $\mathcal{C}(x)$ indicates the color correction matrix at the point x; p_1 and p_2 are the intersection points between the line starting from O and passing through p. These points lie on the acquisition plane immediately before and after p (Figure 2, bottom); d_1, d and d_2 are the distances between the point O and p_1, p and p_2.
Since $C(p_1)$ and $C(p_2)$ are not known in advance, they have to be estimated to evaluate (2). In fact, only few positions on the acquisition plane are measured: hence another interpolation is required. For this planar interpolation we use radial basis method, with gaussians centered on each sample: standard deviation σ defines how much each sample influences the neighbors. In formulas:

$$\mathcal{C}_{ij}(p) = \frac{\sum_{i=1}^{N} \mathcal{C}_{ij}(P_i) \exp\left[-\frac{(p - P_i)^2}{\sigma^2}\right]}{\sum_{i=1}^{N} \exp - \left[\frac{(p - P_i)^2}{\sigma^2}\right]} \tag{3}$$

where N is the total number of samples for the plane and P_i are the positions of samples.
In conclusion, in order to calculate a color correction matrix for a point in the camera space, we first need to calculate two linear interpolations in the acquisition planes, then a polynomial (quadratic) interpolation for the final result.
Regarding the practical implementation, the pre-computed interpolation planes are stored as floating point textures plus, for each texture, some additional information (the distance from the camera center and a scale factor). With this representation, the correction algorithm can be entirely implemented on the GPU, and the pre-computation of the interpolation planes reduces the evaluation of $C_{ij}(p_1)$ and $C_{ij}(p_2)$ in (2) to two texture lookups.

3.3 Analysis of the accuracy of the flash calibration procedure

Before performing a global validation of results on an empirical testbed (see Section 5), we analyzed first the accuracy of the calibration data obtained from the proposed procedure.

A first analysis was performed on the value ranges obtained for the single coefficients of the matrix. Table I shows the statistics relative to all the coefficients calculated for the Nikon camera. The single mean values of each group are very similar, and variance describes a general stability of data. Moreover, color deviation coefficients describe the flash as very near to a white light (with a slight deviation in green channel). Contrast and Intensity offset groups show quite low values in variance, but, as it could be expected, the needed modification of color values increases a lot with the distance.

Further information about the properties of flash light can be inferred from the analysis of the azimuthal and normal sections of the correction space. In Figure 3 the planes associated to coefficient c_{11} are shown: isolines help in understanding the shape of the light. The light wavefront is quite similar for all the models, but

	Contrast			Intensity offset		
Coefficient	c_{11}	c_{22}	c_{33}	c_{14}	c_{24}	c_{34}
Mean value	1.20	1.25	1.33	52.07	50.90	50.47
Variance	0.06	0.09	0.09	81.57	123.5	141.3
Min value	0.83	0.84	0.87	27.5	15.4	14.2
Max value	2.59	2.84	2.86	72.0	75.0	76.8

	Color deviation					
Coefficient	c_{12}	c_{13}	c_{21}	c_{23}	c_{31}	c_{32}
Mean value	0.07	-0.06	0.005	-0.12	0.07	-0.31
Variance	0.001	0.003	0.002	0.009	0.001	0.012
Min value	-0.06	-0.43	-0.29	-0.67	-0.21	-0.85
Max value	0.23	0.01	0.08	0.01	0.12	-0.11

Table I. Statistics of single coefficients for Nikon camera

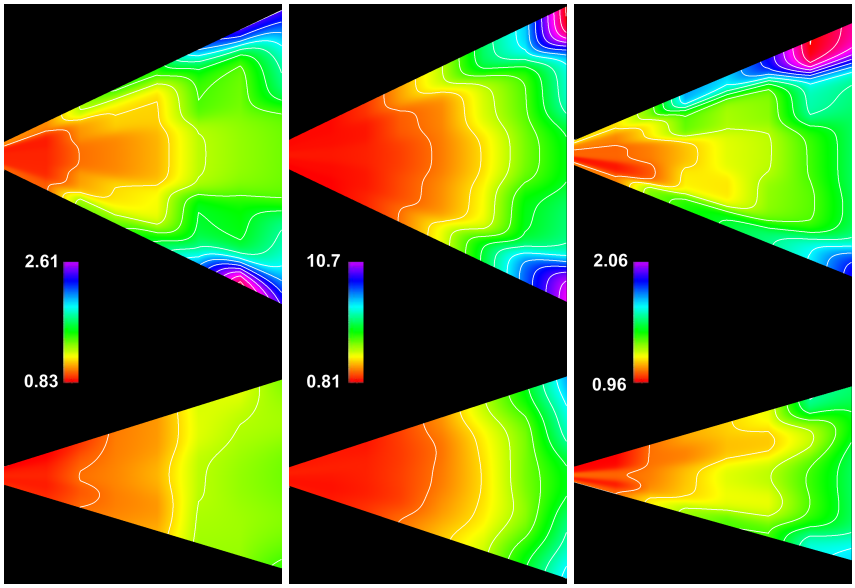

Fig. 3. Azimuthal and normal plane for parameter c_{11} in camera space. (Left) Nikon camera. (Center) Canon camera. (Right) Casio camera.

several differences arise as well. The most regular profile is the one associated to the external flash (Canon camera): this is probably the most reliable kind of illumination. Nevertheless, the isolines show that the shape of the light is not similar to a sphere, but it could be better approximated with series of lobes. Moreover, it can be noted that the maximum value of the coefficient is higher respect to the other two examples: this is because the external flash was set with the largest possible field of view, resulting in a very bright illumination for near points, with the need of more correction for the far ones. The Casio associated space is the less regular one, and this is probably due to the fact that the camera model was the least expensive one, and the images were processed and stored directly in JPEG

format. Anyway, an interesting observation is that the correction space seems to be slightly shifted on the right, like the position of the flash with respect to the lens (Figure 1).

4. REMOVING ILLUMINATION ARTIFACTS (SPECULAR HIGHLIGHTS, SHADOWS)

In order to remove the other artifacts introduced by flash light, we need to carefully estimate the relative position of the flash with respect to the optical sensor of the camera.
When working with built-in flash, the position has to be measured just once, making the process much simpler. Nevertheless, since the camera lens and the flash are very near, our estimation must be accurate to prevent errors in the calculation of reflections and shadows.

While the proximity of the flash and the sensor may suggest that direct physical measurement can produce good results, it is also true that the exact position of the CCD sensor is hidden inside the camera. Moreover, even using a caliper, measuring distances from the inside to the outside of the camera body can be tricky. For this reason we decided to perform an analytical estimation of the flash position.

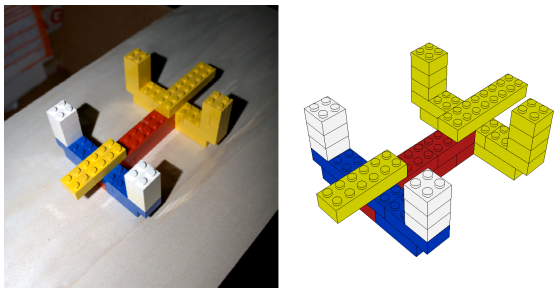

Fig. 4. Left: one of the images used for position estimation; Right, a rendering of the corresponding 3D modeled rig.

4.1 Assessing the accuracy of the light position estimation procedure

There are several approaches to estimate light positions based on either reflection or shadow tracing. We chose a very straightforward and easy-to-implement procedure, which uses one photo and a simple calibration device.

We built a calibration rig using LEGO blocks and modeled the same rig with a 3D modeler [leo]. We took some photos of the device using flash (one example is shown in Figure 4) in order to have shadows over the base plane. We registered the photos using the tool described in [Franken et al. 2005], so that the camera position in space can be computed with sufficient precision. Having the registered image and the 3D model, it was then possible to pick point couples that represented geometric features and their corresponding projected shadows. The picked point couples in 3D space generate a set of lines, whose intersection represents the geometric location of the flash center. The lines intersection point has been calculated as the closest point to all the lines in the set, using the method described in Appendix A. The

Model	X shift	Y shift	Z shift
Casio Exilim Z50	-27 mm	31 mm	1 mm
Nikon D40x	0 mm	75.0 mm	37.0 mm
Canon EOS350D + Flash	0.6 mm	153.2 mm	17.5 mm

Table II. Results of the estimation of the flash position for the three selected camera models.

relative positions of the flash for three cameras (shown in Figure 1-top), obtained with this method, are shown in Table II; these locations are given in a coordinate space centered with the view point of each camera.

The estimated positions proved to be accurate enough to be used for artifacts removal, as shown in Section 5. Direct measurement with a caliper, in the case where it was possible (Nikon SLR camera), gave very similar results, with a 1-2 mm divergence.

4.2 Artifacts removal

Once that the flash position and the color correction space have been reconstructed in the camera calibration phase, we can proceed by detecting and eliminating some of the macroscopic artifacts present in the flash images. As stated in the introduction, we start from the 3D model of the artifact and the set of flash photos which have been registered to the 3D model [Franken et al. 2005]. Provided that the estimation of the reconstructed flash position is correct, there are two artifacts we can recover: highlights and shadows (as presented in the previous section, the single color values in the input images are corrected by calibrating the images with the color correction approach).

4.2.1 *Highlights detection.* Highlights are present on the parts of the 3D surface where specular reflection can happen: specifically, where the ray from the light source would be reflected toward the camera viewpoint. Given the 3D model and the registered image, it is possible to find the highlight areas by using the same realtime technique used to display Phong specular highlights (the half vector technique). Unfortunately, geometric considerations alone are not enough to discriminate highlights in the images, due to local changes of the surface BRDF (that is unknown as well), minor discrepancies of the 3D model w.r.t. the real surface and other similar irregularities. For this reason, we prefer to use this geometric considerations just to select candidates for a possible highlight, and then we decide the actual highlight extent by performing a comparison with the corresponding regions of the other images.

The use of flash light also ensures that the areas of the object which are subject to highlights will be different from one image to the other. This is because the light "follows" the camera in every shot. Hence, using the redundancy between different photos, it is possible to compare the luminance of the candidate point with the luminance of the corresponding area on other photos. The luminance value of the highlight candidate pixel is compared with the average luminance value of the corresponding pixels on the other images. If the difference in luminance is bigger than a fixed threshold, the pixel can be marked as an highlight. We used a two-level threshold: if the luminance value is between 150% and 180% of the average

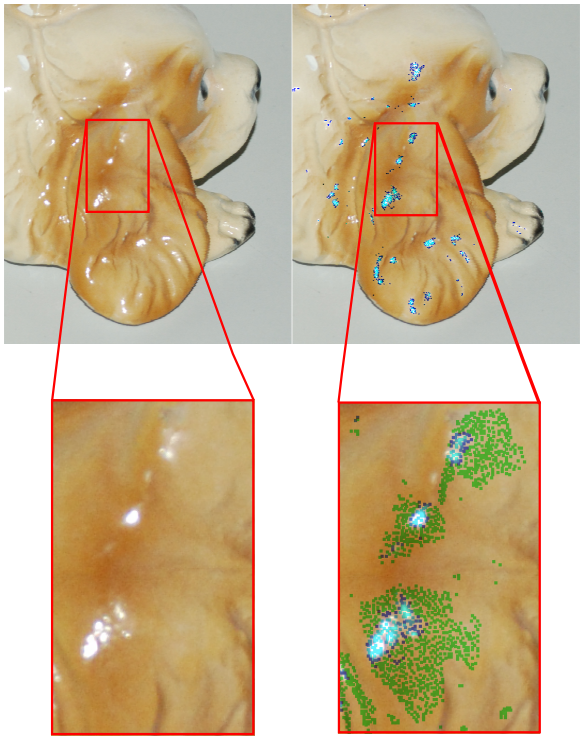

Fig. 5. An example of highlights detection. Upper row: an input flash image and the same image after detection of highlights (blue is the highlight border, cyan is the internal highlight region). Lower row: detail view of a group of highlights with corresponding geometric candidates (in green) and detected highlight pixels.

luminance, the pixel is on the border of the highlight; if it's bigger than 180%, the pixel is considered as completely saturated. This two-levels threshold also reflects the nature of the highlight in low dynamic range images: the border of the highlight (marked in our system with a blending ramp) presents a luminance shift that rises progressively towards the central area, which is composed entirely by over-saturated pixels (marked as completely useless and thus not used in subsequent weighted average computation of final mesh color).

An example of a highlights detection result is shown in Figure 5. The presented model is characterized by a very reflecting material. Most of the highlights are detected automatically (see upper row images). Lower row images present a detail view: given all the pixels mapping on mesh vertices which have been detected as geometric candidates for highlights (green pixels), only a subset is detected as real highlights, and marked as border (blue) or over-saturated (cyan). The green rectangles show two white zones which are small breaks on the real objects. The use of redundancy permits to the system to distinguish between highlights and white colored zones.

4.2.2 *Shadows detection.* Since the flash light is very near to the camera lenses, the amount of shadows in the images is generally low. But due to the nature of this

Fig. 6. An example of shadows detection: left, the original image; right, the shadow detection map.

kind of light, the shadows are very marked and visible. Especially in the context of color projection applications, this results in visible artifacts.
Nevertheless, detecting the parts in the images that are in shadow is even simpler than detecting highlights. Using the camera associated to the specific flash image, it is possible to obtain a depth map for the image. Similarly, given the flash position offset, it is possible to generate the depth map for the light source; comparing the two depth maps, the parts of the flash image which are under shadow are detected. An example of the accurate results obtained for shadows detection is shown in Figure 6. The photo was taken using the Canon camera with external flash, which is positioned to a greater distance from the camera. The shadow position is detected with great accuracy at any distance from the viewpoint. The good results in shadow removal are also an indirect demonstration that the flash position was estimated in a sufficiently accurate way in the camera calibration phase.

5. RESULTS

5.1 Validation of the color correction space

In order to evaluate the results of the correction introduced by the color correction space, we created a setup where the color values in selected regions were known in advance: the scene was formed by a series of identical objects, set at different distances from the camera. The reference objects were simple structures (formed by bricks of different colors) of LEGO DUPLO©, whose size and color are known to be practically identical for all components. The reference RGB color value of each brick was calculated by calibration with the Mini Macbeth. Seven blocks of LEGO© bricks were put in a setting shown in Figure 7(left), and the corresponding 3D model of this configuration was created. This scene has the advantage that, in every single photos, identical elements are present in different parts of the image and at different distances in space. Images were aligned to the 3D model using a registration tool: thus color correction was performed on each pixel which was framing a structure.
One possible criticism to our work could be: would a simple model, for example a point light, be enough to provide a good correction? For this reason, we modeled

the flash as a point light and we corrected the same images using this model. We obtained an acceptable estimation of the intensity of the light, by modeling the light degradation with a quadratic law and taking into account the knowledge of the distance of each pixel in the scene, and its reference color: we used this light estimation to correct the color value. Results of corrections on one of the images are shown in Figure 7(left): the original, the corrected, and the point light corrected images are shown. While in both cases the colors across the scene seem similar, obtained color values are clearly different between the two results. In order to check the accuracy of color correction, we measured the perceptual distance between produced colors and the previously calculated reference colors. We chose the CIE76 Delta E measure, which is the Euclidean distance between the CIELAB values associated to two colors. If Delta E is smaller than one, colors are perceived as identical; for values smaller than eight-ten, the colors can be considered as very similar.

Figure 7 shows the results of the calculation of these distance values. The top

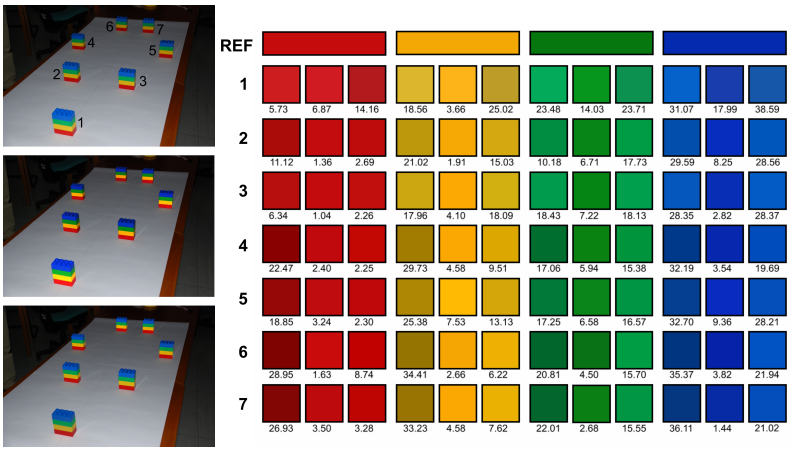

Fig. 7. Left: example of images used for validation: top, original; center, corrected; bottom, point light model corrected. Only the Lego blocks part of the image was corrected. Right: correction accuracy estimation: for each Lego block, average color and Delta E value (respect to reference) in original (left column), corrected (center column) and Point Light corrected (right column) images are shown

boxes display reference colors, then for each piece (each line is associated to the piece number indicated in Figure 7 left) the three columns represent the average color value for original, corrected and point light corrected image. The Delta E value for each color shows the distance with respect to the reference. It is quite clear that, although producing similar colors, the point light correction returns results which are different respect to the reference: only red color is accurately corrected. Slight improvements are introduced only for distant objects. In order to achieve better results, probably a different modeling for each channel would be necessary. On the contrary, our correction proves to be very reliable, with an average Delta E value which is always smaller than ten. Only in the case of block number *1* the correction

is less effective: this can be due to the fact that the object is out of focus, or the color in the original image may be saturated.

A second validation experiment was performed in order to show that the color correction can be reliable also with low quality geometric information of the scene. We reconstructed the geometry of a common scene starting from a few photos, using the Arc3D web service [Vergauwen and Gool 2006]. Figure 8 shows the starting image, together with the geometry obtained, and the corrected image (the correction was applied only where geometry data was present). While the obtained 3D model is only an approximated representation of the geometry in the image, the result of our correction space is satisfying: for example the color of the tablecloth after correction is the same throughout the scene.
Analysis of validation tests brings us to three main conclusions: first of all, color correction is a very reliable way to correct images; moreover, simpler light models are not able to achieve comparable results. Finally, the approach is reliable regardless of the quality of the geometry associated to the image.

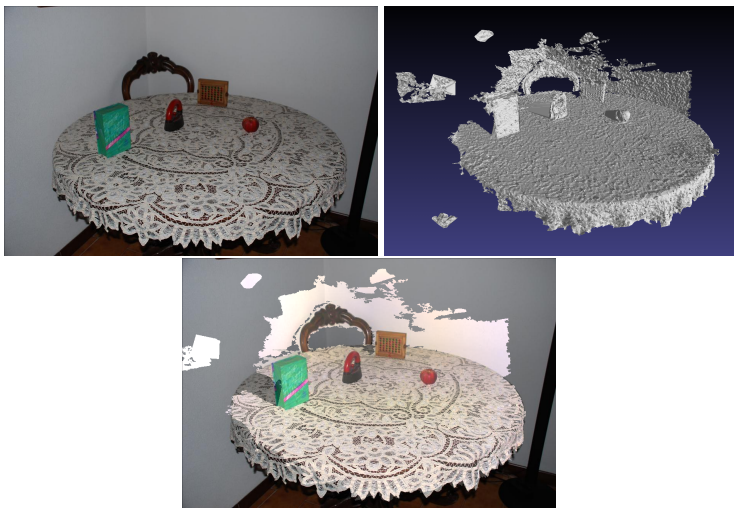

Fig. 8. Example of color correction using low quality geometry: Top left, original flash image; Top right, a snapshot of the extracted geometry; Bottom, corrected image.

5.2 Global assessment over practical applications

The methodologies for spatial color correction and artifact removal presented in the previous sections are quite general, and can be profitably used in different situations. To show the potentiality of this kind of processing, we show its impact in the framework of color mapping from photos. We followed the approach described in [Callieri et al. 2008] where the color assigned to every vertex of the 3D model is a computed as weighted sum of the contributions of all the photos which project on that vertex. These weights are a kind of per-pixel masks that specify importance values that are automatically computed based on several metrics (e.g. distance

from the sensor, camera orthogonality, focusness). The properties of these weights guarantee a smooth blending between photos, without loss of detail; however, the final results can suffer from the fact that the illumination is not known in advance. In particular, this mapping approach is fast, robust and easy to be implemented, but it cannot automatically deal with highlights, hard shadows and strong localized light (as the flash produces), as shown in the examples in Figure 9, where we have undesired lighting artifacts projected on the model. The purpose of the research described in this paper is to find a simple and versatile way to deal with those very artifacts.

Fig. 9. Details of two colored models produced with (no flash images), visualized without illumination and shading (only color values assigned to vertices): multiple highlights show up on the digital models (left); the presence of shadows and different illumination between photos generates a non completely continuous color (right).

We selected a test set of artifacts to assess the quality and the impact of the flash light approach. The test set is a group of objects of different heights (from 20 to 80 cm.), which are characterized by different colors and reflecting materials. We 3D scanned all the objects and acquired photos (from 13 to 32 photos for each object, depending on object size and complexity). The photos were taken turning off the lights in the room, thus having flash light as the principal light. The Nikon D40x was used for most of the test presented here, the other two cameras have also been used to test the accuracy and applicability of the method.

The color mapping approach [Callieri et al. 2008] was easily extended by applying the color correction space before the projection and adding another weighting mask that takes into account the result of artifact removal methods. In particular, the weight of pixels detected as shadows or saturated highlights were assigned to zero, while the pixels detected as borders of the highlights were assigned to a weight value progressively increasing, in order to provide a smooth mask transition.

Figure 10 shows two examples where the highlight removal produced considerable improvement in the final result. In the upper row, several spot-like highlights were removed (one of the images used for projection is the one shown in Figure 5). In the lower row, some more complex in shape highlights were completely removed from a 20 cm Nativity statue.

The effect of shadows removal in most cases appears more subtle with respect to the highlight processing, this because, after the blended mapping, the residual trace

of shadows is just a darkening of areas that can often go unnoticed. However, when the hard shadow line is visible, the advantage of the removal process is significant, as shown in the example in Figure 11, which shows a detail of a model with and without shadows removal. It can be noted that the even very small shadows are detected, such as the shadows projected on the back of the leg and on the top of the foot (see framed regions in the image).

Fig. 10. Details of two colored models, visualized with no illumination (only color values assigned to vertices): on left, the result without highlights removal, on right the model after applying highlights removal.

The effects of color correction can be seen in Figure 12, where the Piper statue is visualized with no illumination. The model on the right, obtained with color correction, appears much more "flat" respect to the model produced without color correction (left), which suffers from the discrepancies between the original images.

The improvements in the obtained results show that flash lighting space sampling can produce better and more faithful results in color projection on 3D models, but also contribute in increasing the performance of the mapping process, since all the artifacts removal methods are completely automatic and it is not necessary for a user to manually mark the artifacts or correct the final result.

These results show that the estimation of the flash light position permits to automatically remove almost all the lighting artifacts from the 3D model. Obviously, like all the contexts where no information about the material is known in advance, some conditions could lead to unsatisfying results. This happens for example when

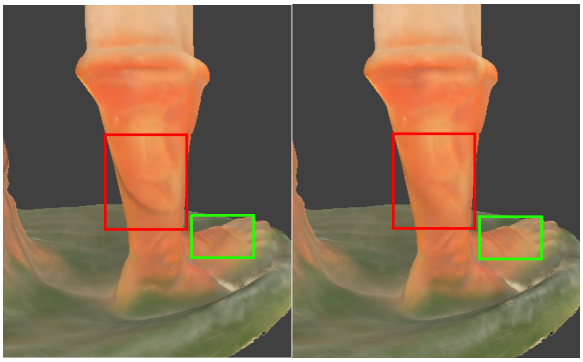

Fig. 11. Details of two colored models, visualized with no illumination (only color values assigned to vertices): on left, the result without shadows removal, on right the model with also shadows removal.

Fig. 12. An example of color correction result, visualized with no illumination (only color values assigned to vertices); color correction is OFF in the left-most image and ON in the right-most one.

there is not enough redundancy in the photo data set, or when the object material presents a peculiar BRDF behavior. One example of the latter is shown in Figure 13: the golden jug in the left-most image presents a metallic-flake paint with unusual reflectance. Both the standard 3D-mapped (center) and the flash-light enhanced (right) reconstructed color result in a not sufficiently realistic output when rendered. In this case, further investigation on the original data, or a user assisted intervention are needed to reconstruct the original color of the model. However, it must be stressed that materials with particular reflectance (e.g. gold patinas) often interfere also with active optical geometry acquisition, so they are rarely considered for 3D scanning using standard acquisition devices.

6. CONCLUSIONS

We presented a novel approach to compute a spatial color correction on images taken using flash light, to compute the relative location of the flash wrt. the camera,

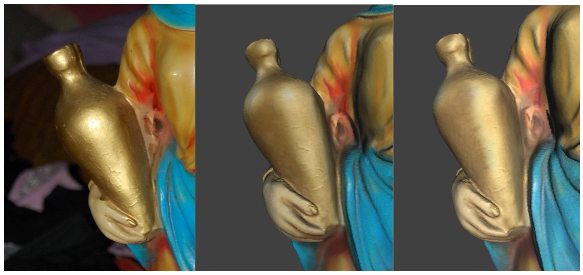

Fig. 13. Left: image of the detail of a Nativity statue (real photo). Center: rendering of the color reconstruction without flash artifacts removal. Right: rendering of the color reconstruction with artifacts removal

and to finally remove usual lighting defects from the textured 3D models produced from those input images and a 3D scanned model. The method presented assumes that some basic information of the geometry of the scene is known in advance (since its main application is on color scanned 3D model, this is not a limitation), and it's able to automatically correct the color values in the image. The proposed flash calibration procedure needs to be estimated only once in a camera lifetime (assuming that the flash will not change its behavior during the life span of the device) and this calibration procedure is adequately fast and easy. The proposed method has several advantages: it is robust and flexible, and although the calibration procedure best fits with the case of flash light, the approach presented can be used with any kind of light which is spatially bound to the camera.

Moreover, we also presented a simple and automatic procedure able to identify and remove other annoying artifacts, like highlight and shadows. This enhancement is allowed by the knowledge of the flash calibration data (for each image, we know the major lighting source location and incidence direction on each surface parcel sampled by each pixel).

Future work will include both improvements of the proposed approach and exploitation of other possible uses.
Regarding the first issue, further efforts could be put in trying to find an even simpler acquisition procedures, or to reduce the number of samples positions.
Focusing on other possible future applications, we can outline three possible uses for color correction:
Image enhancement: the concept of image enhancement is not necessarily related to a complete color correction. It could be possible to interactively apply color correction by using part the elements of the matrix (i.e. correcting color deviation only) to obtain the desired effect, which could be either an improved visual pleasantness or an enhanced readability.
Appearance Acquisition: another potential application of our method is in the framework of the capture of the appearance properties of a material. Most estimation approaches rely on controlled illumination mechanical setup to move the object or the light. By working on different flash-lighted images and extracting the local illumination data from the the correction space, it should be possible to recover at least an approximation of the surface's optical properties.

Flash Modeling: the spatial intensity variation recovered from our structure could allow the definition of mathematical model of the flash light. While the simple point light model exhibits strong limitations, it could be possible to define a more complex model that could approximate in a very accurate way the behavior of the flash light.

Appendix A: How to find the closest point to N lines

With closest point to a given set of lines we intend the point having the minimum Euclidean distance with respect to those lines. Typically, this problem is formulated using Plücker coordinates. Instead, here we compute this point by solving the problem in a closed form, since the resulting matrices are not ill-conditioned in our case. More precisely, by indicating the set of n lines with

$$L = \left\{ l_i = O_i + t\vec{d_i} | \, t \in \mathbb{R} \right\} \quad i = 1 \dots n \tag{4}$$

where O_i is the origin of the i-th line and $\vec{d_i}$ is the corresponding direction (normalized), we found the closest point by minimizing:

$$p = \arg\min_x \sum_{i=1}^{n} d(x, l_i) \tag{5}$$

The distance $d(x, l_i)$ can be written as

$$d(x, l_i)^2 = (x - O_i) \left[\mathbf{I} - \vec{d_i}\vec{d_i}^T \right] (x - O_i) \tag{6}$$

The minimization is obtained by substituting (6) in (5), and imposing the derivative to zero. After some simple algebra we obtain the final formulation:

$$p = \left[n\mathbf{I} - \sum_{i=1}^{n} \vec{d_i}\vec{d_i}^T \right]^{-1} \sum_{i=1}^{n} \left[\mathbf{I} - \vec{d_i}\vec{d_i}^T \right] O_i \tag{7}$$

Acknowledgment

We acknowledge the financial support of the EC IST IP project "3D-COFORM" (IST-2008-231809).

REFERENCES

BT Software.
`http://www.leocad.org/`.

Agrawal, A., Raskar, R., Nayar, S. K., and Li, Y. 2005. Removing photography artifacts using gradient projection and flash-exposure sampling. *ACM Tr. on Graphics 24,* 3, 828–835.

Bannai, N., Agathos, A., and Fisher, R. B. 2004. Fusing multiple color images for texturing models. In *3DPVT '04: Proceedings of the 3D Data Processing, Visualization, and Transmission, 2nd International Symposium.* IEEE Computer Society, Washington, DC, USA, 558–565.

Barnard, K., Cardei, V., and Funt, B. 2002. A comparison of computational color constancy algorithms. Part I: Methodology and experiments with synthesized data. *Image Processing, IEEE Trans. 11,* 9, 972–984.

Barnard, K., Martin, L., Coath, A., and Funt, B. 2002. A Comparison of Computational Color Constancy Algorithms Part II: Experiments With Image Data. *Image processing, IEEE Trans. on 11,* 9, 985.

Baumberg, A. 2002. Blending images for texturing 3d models. In *BMVC 2002*. Canon Research Center Europe.

Bernardini, F., Martin, I., and Rushmeier, H. 2001. High-quality texture reconstruction from multiple scans. *IEEE Transactions on Visualization and Computer Graphics 7,* 4, 318–332.

Callieri, M., Cignoni, P., Corsini, M., and Scopigno, R. 2008. Masked photo blending: mapping dense photographic dataset on high-resolution 3d models. *Computer & Graphics 32,* 4 (Aug), 464–473. for the online version: http://dx.doi.org/10.1016/j.cag.2008.05.004.

Callieri, M., Cignoni, P., and Scopigno, R. 2002. Reconstructing textured meshes from multiple range rgb maps. In *7th Int.l Fall Workshop on Vision, Modeling, and Visualization 2002*. IOS Press, Erlangen (D), 419–426.

Commission Internationale de l'Eclairage (CIE) 2004. *Colorimetry. CIE 15:2004*. Commission Internationale de l'Eclairage (CIE).

Corsini, M., Callieri, M., and Cignoni, P. 2008. Stereo light probe. *Computer Graphics Forum 27,* 2, 291–300.

Debevec, P. 1998. Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*. ACM Press, New York, NY, USA, 189–198.

Debevec, P., Hawkins, T., Tchou, C., Duiker, H.-P., Sarokin, W., and Sagar, M. 2000. Acquiring the reflectance field of a human face. In *SIGGRAPH 2000*. ACM Press/Addison-Wesley, New York, NY, USA, 145–156.

Ebner, M. 2004. Color constancy using local color shifts. In *ECCV*. 276–287.

Eisemann, E. and Durand, F. 2004. Flash photography enhancement via intrinsic relighting. In *ACM Trans. on Graphics*. Vol. 23. ACM Press.

Finlayson, G. D. and Hordley, S. D. 2000. Improving gamut mapping color constancy. *Image Processing,IEEE Trans. 9,* 10, 1774–1783.

Finlayson, G. D., Hordley, S. D., and Hubel, P. M. 2001. Color by correlation: A simple, unifying framework for color constancy. *IEEE Trans. on Pattern Analysis and Machine Intelligence 23,* 11, 1209–1221.

Franken, T., Dellepiane, M., Ganovelli, F., Cignoni, P., Montani, C., and Scopigno, R. 2005. Minimizing user intervention in registering 2d images to 3d models. *The Visual Computer 21,* 8-10 (sep), 619–628. Special Issues for Pacific Graphics 2005.

Gijsenij, A. and Gevers, T. 2007. Color constancy using natural image statistics. In *Int. Conf. on Comp. Vision and Pat. Recogn.* Minneapolis, USA, 1–8.

Grubbs, F. 1969. Procedures for detecting outlying observations in samples. *Technometrics 11*, 1–21.

Hoppe, H. and Toyama, K. 2003. Continuous flash. Tech. Rep. MSR-TR-2003-63, Microsoft Research.

Hsu, E., Mertens, T., Paris, S., Avidan, S., and Durand, F. 2008. Light mixture estimation for spatially varying white balance. In *SIGGRAPH '08*. ACM Press.

Kawakami, R., Ikeuchi, K., and Tan, R. T. 2005. Consistent surface color for texturing large objects in outdoor scenes. In *ICCV '05: Int. Conf. on Computer Vision*. Washington, DC, USA, 1200–1207.

Lensch, H., Heidrich, W., and Seidel, H. 2000. Automated texture registration and stitching for real world models. In *8th Pacific Graphics Conference*. IEEE, Los Alamitos, CA, 317–327.

Lensch, H. P. A., Kautz, J., Goesele, M., Heidrich, W., and Seidel, H.-P. 2003. Image-based reconstruction of spatial appearance and geometric detail. *ACM Trans. Graph. 22,* 2, 234–257.

Lin, S., Yuanzhen, S. L., Kang, S. B., Tong, X., and yeung Shum, H. 2003. Diffuse-specular separation and depth recovery from image sequences. In *In Proceedings of European Conference on Computer Vision (ECCV)*. 210–224.

Lischinski, D., Farbman, Z., Uyttendaele, M., and Szeliski, R. 2006. Interactive local adjustment of tonal values. *ACM Tr. Gr. 25,* 3, 646–653.

Lu, C., Drew, M. S., and Finlayson, G. D. 2006. Shadow removal via flash/noflash illumination. *W. on Mult. Signal Processing*, 198–201.

Omer, I. and Werman, M. 2004. Color lines: Image specific color representation. In *CVPR 04*. Vol. II. IEEE, 946–953.

Ortiz, F. and Torres, F. 2006. Automatic detection and elimination of specular reflectance in color images by means of ms diagram and vector connected filters. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 36,* 5 (Sept.), 681–687.

Petschnigg, G., Szeliski, R., Agrawala, M., Cohen, M., Hoppe, H., and Toyama, K. 2004. Digital photography with flash and no-flash image pairs. *SIGGRAPH '04*, 664–72.

Pulli, K., Abi-Rached, H., Duchamp, T., Shapiro, L. G., and Stuetzle, W. 1998. Acquisition and visualization of colored 3d objects. In *Proceedings of ICPR 98*. IEEE Computer Society, Washington, DC, USA, 11.

Rankov, V., Locke, R., Edens, R., Barber, P., and Vojnovic, B. 2005. An algorithm for image stitching and blending. In *Proceedings of SPIE. Three-Dimensional and Multidimensional Microscopy: Image Acquisition and Processing XII*. Vol. 5701. 190–199.

Raskar, R., Welch, G., Low, K., and Bandyopadhyay, B. 2001. Shader lamps: Animating real objects with image-based illumination. In *Rendering Techniques 2001, The Eurographics Workshop on Rendering*. Springer-Verlag, 89–102.

Sato, Y. and Ikeuchi, K. 1993. Temporal-color space analysis of reflection. In *Computer Vision and Pattern Recognition*. 570–576.

Shen, H.-L., Zhang, H.-G., Shao, S.-J., and Xin, J. H. 2008. Chromaticity-based separation of reflection components in a single image. *Pattern Recogn. 41,* 8, 2461–2469.

Tan, P., Lin, S., Quan, L., and Shum, H.-Y. 2003. Highlight removal by illumination-constrained inpainting. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*. IEEE Computer Society, Washington, DC, USA, 164.

van de Weijer, J. and Gevers, T. 2005. Color constancy based on the grey-edge hypothesis. In *ICIP* (2007-05-04). 722–725.

Vergauwen, M. and Gool, L. V. 2006. Web-based 3d reconstruction service. *Mach. Vision Appl. 17,* 6, 411–426.

Wolff, L. 1989. Using polarization to separate reflection components. In *Computer Vision and Pattern Recognition, 1989*. 363–369.

...