

# On-the-go Reflectance Transformation Imaging with Ordinary Smartphones

Mara Pistellato<sup>1</sup>[0000-0001-6273-290X]  
and Filippo Bergamasco<sup>1</sup>[0000-0001-6668-1556]

DAIS, Università Ca'Foscari Venezia  
155, via Torino, Venezia Italy  
{mara.pistellato,filippo.bergamasco}@unive.it

**Abstract.** Reflectance Transformation Imaging (RTI) is a popular technique that allows the recovery of per-pixel reflectance information by capturing an object under different light conditions. This can be later used to reveal surface details and interactively relight the subject. Such process, however, typically requires dedicated hardware setups to recover the light direction from multiple locations, making the process tedious when performed outside the lab.

We propose a novel RTI method that can be carried out by recording videos with two ordinary smartphones. The flash led-light of one device is used to illuminate the subject while the other captures the reflectance. Since the led is mounted close to the camera lenses, we can infer the light direction for thousands of images by freely moving the illuminating device while observing a fiducial marker surrounding the subject. To deal with such amount of data, we propose a neural relighting model that reconstructs object appearance for arbitrary light directions from extremely compact reflectance distribution data compressed via Principal Components Analysis (PCA). Experiments shows that the proposed technique can be easily performed on the field with a resulting RTI model that can outperform state-of-the-art approaches involving dedicated hardware setups.

**Keywords:** Reflectance Transformation Imaging; Neural Network; Camera Pose Estimation; Interactive Relighting

## 1 Introduction

In Reflectance Transformation Imaging (RTI) an object is acquired with different known light conditions to approximate the per-pixel Bi-directional Reflectance Distribution Function (BRDF) from a static viewpoint. Such process is commonly used to produce relightable images for Cultural Heritage applications [19, 6] or perform material quality analysis [4] and surface normal reconstruction. The flexibility of such method makes it suitable for several materials, and the resulting images can unravel novel information about the object under study such as manufacturing techniques, surface conditions or conservation treatments. Among the variety of practical applications in Cultural Heritage field, we can

mention enhanced visualisation [6, 21], documentation and preservation [16, 13, 15] as well as surface analysis [3]. Moreover, RTI techniques can be effectively paired with other tools as 3D reconstruction [36, 23–25] or multispectral imaging [8] to further improve the results.

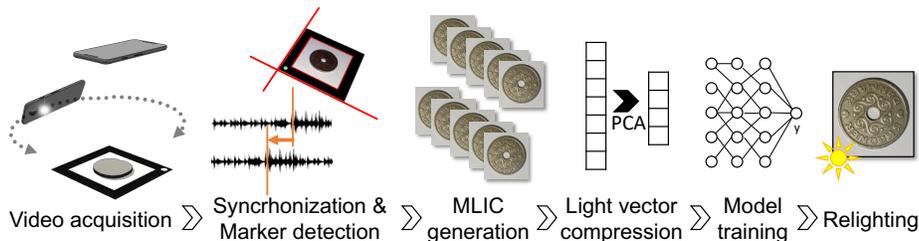
In the majority of the cases, the acquisition of RTI data is carried out with specialised hardware involving a light dome and other custom devices that need complex initial calibration. Since the amount of processed data is significant, several compression methods have been proposed for RTI data representation to obtain efficient storage and interactive rendering [27, 9]. In addition to that, part of the proposals focus on the need of low-cost portable solutions [12, 38, 28], including mobile devices [31] to perform the computation on the field.

In this paper we first propose a low-cost acquisition pipeline that requires a couple of ordinary smartphones and a simple marker printed on a flat surface. During the process, both smartphones acquire two videos simultaneously: one device acting as a static camera observing the object from a fixed viewpoint, while the other provides a trackable moving light source. The two videos are synchronised and then the marker is used to recover the light position with respect to a common reference frame, originating a sequence of intensity images paired with light directions. The second contribution of our work is an efficient and accurate neural-network model to describe per-pixel reflectance based on PCA-compressed intensity data. We tested the proposed relighting approach both on a synthetic RTI dataset, involving different surfaces and materials, and on several real-world objects acquired on the field.

## 2 Related Work

The literature counts a huge number of different methods for both acquisition and processing of RTI data for relighting. In [22] the authors give a comprehensive survey on Multi-Light Image Collections (MLICs) for surface analysis. Many approaches employ the classical polynomial texture maps [14] to (i) define the per-pixel light function, (ii) store a representation of the acquire data, and (iii) dynamically render the image under new lights. Similar techniques are the so-called Hemispherical Harmonics coefficients [17] and Discrete Modal Decomposition [26]. In [9] the authors propose a new method based on Radial Basis Function (RBF) interpolation, while in [27] a compact representation for web visualisation employing PCA is presented. The authors in [18] present the High-light Reflectance Transformation Imaging (H-RTI) framework, where the light direction is estimated by detecting its specular reflection on one or more spherical objects captured in the scene. However, such setup involves several assumptions such as constant light intensity and orthographic camera model, that in practice make the model unstable. Other techniques that have been proposed to estimate light directly from some scene features are [1, 2], while in the authors [9] propose a novel framework to expand the H-RTI technique.

Recently, neural networks have been employed successfully in several Computer Vision tasks, including RTI. In particular, the encoder-decoder architec-



**Fig. 1.** Complete mobile-based RTI acquisition and relighting pipeline.

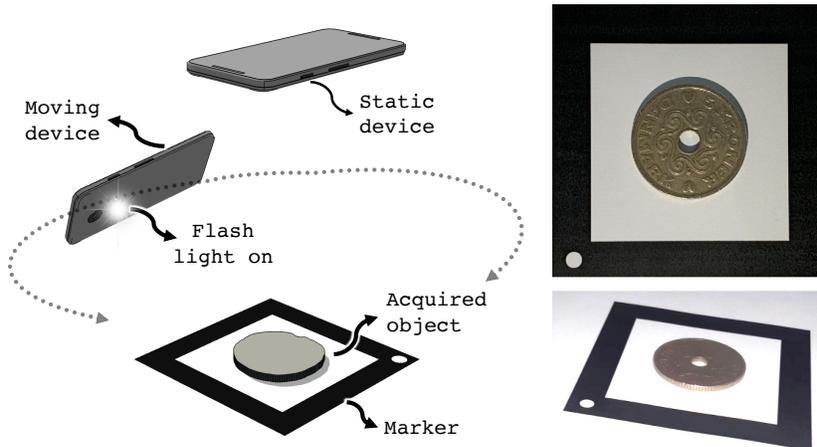
ture is used in several applications for effective data compression [33]. The work in [30] presents a NN-based method to model light transport as a non-linear function of light position and pixel coordinates to perform image relighting. Other related work using neural networks are [39], in which a subset of optimal light directions is selected, and [29] where a convolutional approach is adopted. Authors in [5] propose an autoencoder architecture to perform relighting of RTI data: the architecture is composed by an encoder part where pixel-wise acquired values are compressed, then the decoder part uses the light information to output the expected pixel value. They also propose two benchmark datasets for evaluation.

### 3 Proposed Method

Our method follows the classical procedure employed in the vast majority of existing RTI applications: the whole pipeline is presented in Figure 1. First, several images of the object under study are acquired varying the lighting conditions. In our case, the operation uses the on-board cameras and flash light of a pair of ordinary smartphones while taking two videos. The two videos are then synchronised and the smartphones positions with respect to the scene are recovered using a fiducial marker: in this way we obtain light position and reflectance image for each frame. Such data is processed to create a model that maps each pair (*pixel, light direction*) to an observed reflectance value. Section 3.1 gives a detailed description of this process. This results in a Multi-Light Image Collection (MLIC), that is efficiently compressed by projecting light vectors to a lower-dimensional space via PCA. Then, we designed a neural model defined as a small Multi-Layer Perceptron (MLP) to decode the compressed light vectors and extrapolate the expected intensity of a pixel given a light direction. In Section 3.2 the neural reflectance model and data compression are illustrated in detail. Finally, the trained model is used to dynamically relight the object by setting the light direction to any (possibly unseen) value.

#### 3.1 Data Acquisition

Data acquisition is performed using two smartphones and a custom fiducial marker as shown in Figure 2 (left). The object to acquire is placed at the centre of a marker composed by a thick black square with a white dot at one corner.



**Fig. 2.** Left: Proposed RTI acquisition setup. Right: Example frames acquired by the static and moving devices.

One device is located above the object, with the camera facing it frontally so that it produces images as depicted in Figure 2 (top-right). This device, called *static*, must not move throughout the acquisition, so we suggest to attach it to a tripod. The second device, called *moving*, is manually moved around the object with an orbiting trajectory. The flash led-light located close to the backward-facing camera must be kept on all the time to illuminate the object from different locations. This will allow the static device to observe how the reflectance of each pixel changes while moving the light source.

Both the devices record a video during the acquisition. For now, let's consider those videos as just sequences of images perfectly synchronised in time. In other words, the acquisition consists in a sequence of  $M$  images  $(I_0^s, I_1^s, \dots, I_M^s)$  acquired from the static device paired with a sequence  $(I_0^m, I_1^m, \dots, I_M^m)$  acquired from the moving device at times  $t_0, t_1, \dots, t_M$ .

After video acquisition, each image is processed to detect the fiducial marker. For the static camera, this operation is needed to locate the 4 corners  $(c_0, c_1, c_2, c_3)$  of the inner white square (i.e. the internal part of the marker inside the thick black border). This region is then cropped to create a sequence of  $(\mathcal{I}_0, \dots, \mathcal{I}_N)$  images composed by  $W \times H$  pixels commonly referred as Multi-light Image Collection (MLIC). Note that  $N$  can be lower than  $M$  because the fiducial marker must be detected in both  $I_i^s$  and  $I_i^m$  to be added to the MLIC.

Each  $\mathcal{I}_i$  is a single-channel grayscale image containing only the luminance of the original  $I_i^s$  image. We decided to model only the reflectance intensity (and not the wavelength) as a function of the light's angle of incidence for two reasons. First, we cannot change the colour of the light source and, second, it is uncommon to have iridescent materials where the incident angle affects the reflectance spectrum [11]. Therefore, we convert all the images to the YUV colour

space to store only the Y channel in the MLIC. To deal with the colour, we store the pixel-wise averages  $\bar{U} = \frac{1}{N} \sum_N U_i$  and  $\bar{V} = \frac{1}{N} \sum_N V_i$  for further processing.

The marker is also detected in the moving camera image sequence, but for a different purpose. We assume that the flash light is so close to the camera optical centre that can be considered almost at the same point. So, by finding the pose of the camera  $(R, t)$  in the marker reference frame, we can estimate the location of the light source (i.e. the moving camera optical center) with respect to the object. This operation is simply performed by computing the Homography  $H$  mapping  $c_0 \dots c_3$  to the marker model points  $\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} W \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} W \\ H \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ H \\ 1 \end{pmatrix}$  and then factorising it as:

$$K^{-1}H = \alpha \begin{pmatrix} | & | & | \\ r_1 & r_2 & t \\ | & | & | \end{pmatrix} \quad (1)$$

where  $K$  is the intrinsic camera matrix,  $r_1, r_2$  are the first two columns of the rotation matrix  $R$ , and  $\alpha$  is a non-zero unknown scale factor [10]. Since  $R$  must be orthonormal,  $\alpha$  can be approximated as  $2/(\|r_1\| + \|r_2\|)$  and  $r_3$  as  $r_1 \times r_2$ . Since  $(R, t)$  maps points from the camera reference frame to the marker (i.e. object) reference frame, the vector  $t$  represents the light position and  $\mathcal{L} = t/\|t\|$  the light direction. Since the light is not at the infinity, each object point actually observes a slightly different light direction vector  $\mathcal{L}$ . However, as usually done in other RTI applications, we consider this difference negligible so that we collect a single light direction vector for each image.

After the data acquisition process, we end up with the MLIC  $(\mathcal{I}_0 \dots \mathcal{I}_N)$  and  $(\mathcal{L}_0, \dots, \mathcal{L}_N)$  vectors together with  $\bar{U}, \bar{V}$ . This is all the data we need to generate our reflectance model and proceed with dynamic relighting. At this point, we need to do some considerations regarding the acquisition procedure:

- Pixel reflectance data is collected only from the static camera, while the moving camera is used just to estimate the light direction. This implies that the final result quality is directly affected by the quality of the static camera (i.e. resolution, noise, etc.). Therefore, we suggest to use a good smartphone for that. Conversely, the moving device can be cheap as long as images are sufficiently well exposed to reliably detect the marker.
- The moving camera must be calibrated a-priori to factorise  $H$ . In practice, the calibration is not critical and can simply be inferred from the lens information provided in the EXIF metadata. We also used this approach in all our experiments.
- The ambient should be illuminated uniformly and constantly over time. Ideally, the moving device flash light should be the only one observed by the object. Since that is typically impractical, it is at least sufficient that the contribution from ambient illumination is negligible with respect to the provided moving light.

- The orbital motion should uniformly span the top hemisphere above the object with a certain constant radius. Indeed, we only consider light direction so changes in the reflectance due to light proximity with respect to the object will not be properly accounted by the model.

**Video Synchronisation** Video recording is manually started (roughly at the same time) in the two devices. So, it is clear that the two frame sequences are not synchronised out of the box (i.e. the  $i^{th}$  frame of the static device will not be taken at the same time as the  $i^{th}$  frame of the moving one). That will never be the case without an external electronic triggering but we can still obtain a reasonable synchronisation exploiting the audio signal of the two videos [37].

We first extract the two audio tracks and then compute the time offset in seconds that maximises the Time-lagged Cross-correlation [32]. Once the offset is known, initial frames from the video starting first are dropped to match the two sequences. Note also that, if the framerates are different, frames must be dropped from time to time from the fastest video to keep it in sync with the other. In the worst case, the time skewness is  $1/\text{FPS}$  where FPS is the framerate of the slowest video. Nevertheless, since the moving device is orbited around the object very slowly, such time skewness will have a negligible effect in the estimation of  $(\mathcal{L}_0, \dots, \mathcal{L}_N)$ .

**Fiducial Marker Detection** Detecting the four internal corners of the proposed fiducial marker can be simply performed with classical image processing. We start with Otsu’s image thresholding [20] followed by hierarchical closed contour extraction [34]. Each contour is then simplified with the Ramer-Douglas-Peucker algorithm and filtered out if resulting in a number of points different than 4. All the black-to-white 4-sides polygons contained into a white-to-black 4-side polygon are good candidates for a marker. So, we check the midpoint of each closest corresponding vertex pairs searching for the white dot. If exactly one white dot is found among corresponding pairs, than the four vertexes of the internal polygon can be arranged in clockwise order starting from the one closest to the dot. This results in the four corners  $c_0, \dots, c_3$ .

Since we expect to see exactly one instance of the marker in every frame, this simple approach is sufficient in practice. We decided not use popular alternative markers (see for instance [7]) because they typically reserve the internal payload area to encode the marker id. Of course, any method will work as long as it results in a reasonably accurate localisation of the camera while providing free space to place the object under study.

### 3.2 The Reflectance Model

To perform interactive relighting, we need first to model how the reflectance changes when varying the light direction. Our goal is to define a function:

$$f(\mathbf{p}, \vec{l}) \rightarrow (y, u, v) \quad (2)$$

producing the intensity  $y$  and colour  $u, v$  (in the YUV space) of a pixel  $\mathbf{p} = (x, y) \in \mathbb{N} \times \mathbb{N}$  when illuminated from a light source with direction  $\vec{l} = (l_u, l_v)$ <sup>1</sup>. Once the model is known, relighting can be done by choosing a light  $\vec{l}$  and evaluating  $f$  for every pixel  $\mathbf{p}$  of the target image.

The data acquisition process described before produces a sampling of  $f$  for some discrete values of  $\mathbf{p}$  and  $\vec{l}$ . This sampling is dense on the pixels, since we acquire an entire image for every light, but typically sparse in the amount of the observed light directions, especially if using a light dome where this number is limited to a few dozens. In our case, the sampling of  $\vec{l}$  is a lot denser since we acquire an entire video composed by thousands of frames. However, directions are highly correlated in space as we follow a continuous circular trajectory (See Figure 3, Left).

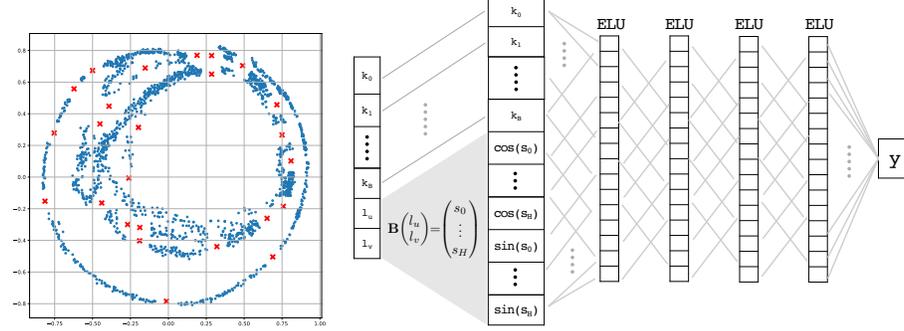
The challenge is to: (i) provide a realistic approximation of  $f$  for previously unseen light directions while (ii) using a very compact representation so that it can be easily transferred, stored and evaluated even on a mobile phone. The two problems are related because the selection of what family of functions to use for  $f$  affects how many parameters are needed to describe the chosen one. For example, in [14] each pixel is independently modelled as a 6-coefficients biquadratic function of the light direction, requiring the storage of  $6 \times M \times N$  values.

Inspired by *NeuralRTI* proposed by Dulecha et. al. [5], we also represent  $f$  as a Multi-Layer Perceptron trained from the data acquired with the smartphones. However, we have two substantial differences with respect to their approach. First, we avoid the auto-encoder architecture for data compression. Since we do not use a light dome, the number of light samples changes in each acquisition and is at least an order of magnitude greater. *NeuralRTI* would not be feasible in our case, as it results in a network taking in input vectors of thousands of elements. Also, the network architecture itself depends on  $N$  (variable in our case) producing a different layout for each acquired object. Instead, we compress such vectors with classical PCA to feed the MLP acting as a decoder. Interestingly, this tend to produce better results not only on our data but also on images acquired with a classical light dome. Second, the light vector  $\vec{l}$  is not concatenated as-is to the network input but projected to a higher-dimensional Fourier space with random frequencies as discussed in [35]. This has a positive effect on the ability of the network to reconstruct the correct pixel intensity.

### 3.3 Neural Model

Our proposed neural model  $\mathcal{Z}(\mathbf{k}_p, \vec{l}) \rightarrow y$  works independently for each pixel (i.e. it does not consider the spatial relationship among those) and recovers the intensity information  $y$  without the colour. It takes as input a compressed light vector  $\mathbf{k}_p = (k_0, k_1, \dots, k_B) \in \mathbb{R}^B$  of any pixel  $p$  and a light direction  $\vec{l}$  to produce the intensity for pixel  $p$ .

<sup>1</sup>  $l_u$  and  $l_v$  range between  $[-1 \dots 1]$  respectively as they are the first two components of a (unitary-norm) 3D light direction vector pointing toward the light source.



**Fig. 3.** Left: 2D plot of the first two components of the light direction vectors ( $\mathcal{L}_0 \dots \mathcal{L}_n$ ). Each point is associated to an image in the MLIC. Note the circular trajectory. Right: Network architecture composed by 5 fully-connected layers with ELU activation.

The model  $\mathcal{Z}$  is composed by an initial (non-trainable) projection of the light vector followed by a MLP arranged in 5 layers consisting in 16 neurons each, all using the ELU activation function except for the output realised with a single neuron with linear activation (Fig. 3, right). The network input  $I$  is a  $(B + 2H)$ -dimensional vector created by concatenating the  $B$  values of  $\mathbf{k}_p$  with the projection of  $\vec{l}$  to an  $H$ -dimensional Fourier space with random frequencies.

Specifically, let  $\mathbf{B}$  be a  $H \times 2$  matrix where each element is sampled from a Gaussian distribution  $\mathcal{N}(0, \sigma^2)$ . This matrix is generated once for each acquired object, it is not trained, and is common to all the pixels. The network input is then obtained as:

$$I = (k_0, \dots, k_B, \cos(s_0), \dots, \cos(s_H), \sin(s_0) \dots \sin(s_H)) \quad (3)$$

$$\text{where } (s_0 \dots s_H)^T = \mathbf{B} \begin{pmatrix} l_u \\ l_v \end{pmatrix}. \quad (4)$$

Once  $\mathcal{Z}$  is trained, it can be used for relighting as follows:

$$f(\mathbf{p}, \vec{l}) = \left( \mathcal{Z}(\mathbf{k}_p, \vec{l}), \bar{U}(\mathbf{p}), \bar{V}(\mathbf{p}) \right). \quad (5)$$

**Creating the compressed light vectors  $\mathbf{k}_p$**  The size of the neural model  $\mathcal{Z}$  depends by the  $2H$  values of the matrix  $\mathbf{B}$ , its internal weights, and the light vectors  $\mathbf{k}_p$  (one for each pixel, for a total of  $W \cdot H \cdot B$  values). It is obvious that most of the storage is spent for the light vectors since the number of image pixels is far greater than the other variables. Considering that we acquire roughly 1 minute of video at 30 FPS, our MLIC is composed by  $\approx 2000$  images cropped to a size of  $400 \times 400$  pixels for a total of 320 MB. So, using all the acquired data as-is to define  $f$  jeopardises the idea of doing interactive relighting directly on a mobile

app or in a web browser. Since we assume that all the pixels observe the same light vector, the acquired MLIC ( $\mathcal{I}_0 \dots \mathcal{I}_N$ ) can be represented as a  $W \times H \times N$   $N$ -channel image in which, for each pixel, a vector of  $N$  values (corresponding to light directions  $\mathcal{L}_0 \dots \mathcal{L}_N$ ) have been observed. In [5] the authors use an auto-encoder to produce an intermediate encoded representation of the observed light vectors of each pixel, and then just the decoder for relighting. This works well for the light dome in which  $N$  is typically less than 50. We tried their approach with  $N = 1500$  lights and realised that the network struggles to converge to an effective encoded representation.

We propose a more classical approach in which the encoding of the light vectors is not based on Deep Learning. Let  $\mathbf{K}_p$  be the  $N$ -dimensional light vector of the pixel  $p$ . We propose to use Principal Component Analysis on all the light vectors acquired  $\mathbf{K}_{p_0} \dots \mathbf{K}_{p_{W \times H}}$  to find a lower-dimensional space of  $B$  orthogonal bases. Then, the encoded  $\mathbf{k}_p$  is obtained by projecting  $\mathbf{K}_p$  into that space. In the experimental section we show how the number of bases  $B$  can be very small compared to  $N$  while still producing high-quality results.

**Network training and implementation details** The network model  $\mathcal{Z}$  is trained by associating each input  $I$  with the expected reflectance intensity. Specifically, we combined the encoded vector  $\mathbf{k}_p$  of each pixel, with all the possible light directions  $\mathcal{L}_0 \dots \mathcal{L}_N$  to produce the input  $I_{x,y,n}$  ( $0 \leq y < H$ ,  $0 \leq x < W$ ,  $0 \leq n < N$ ). The output associated to  $I_{x,y,n}$  is simply the value of  $\mathcal{I}_n(x, y)$ , that is the intensity value observed for light  $n$  at pixel  $(x, y)$ . This results in a total of  $W \cdot H \cdot N$  data samples to be used for training. Note that, regardless the amount of pixels (i.e. the image resolution) and light directions involved (i.e. number of frames in the video), the network architecture remains unchanged. Therefore, it is easy to compute how much storage is needed for the model, depending only on the number of PCA bases  $B$  and image resolution. Considering the acquired data size discussed before, and supposing to use  $B = 8$  bases and  $H = 10$  frequencies, we have to store  $400 \times 400 \times 8$  compressed light vectors ( $\approx 5$  MB with single precision), and 1252 values for network weights and  $\mathbf{B}$ .

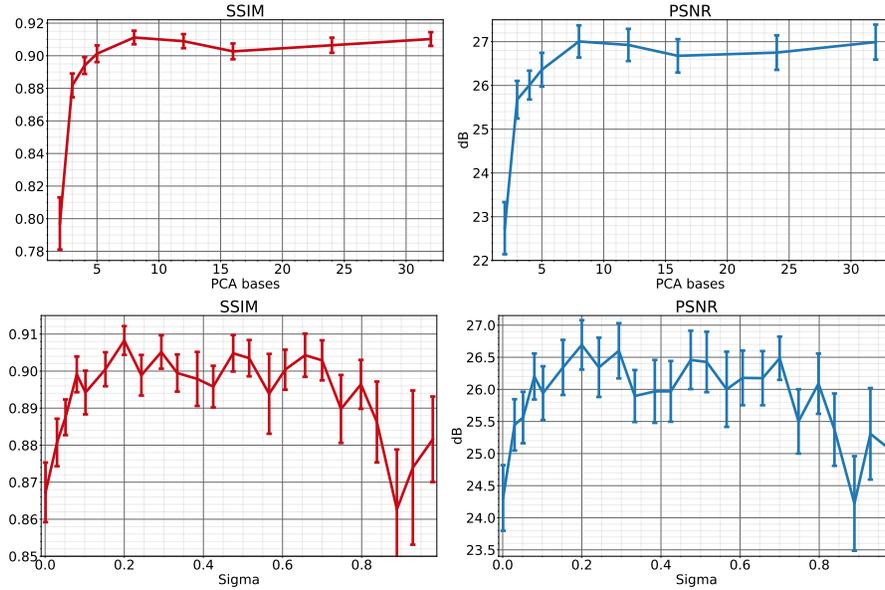
Finally, we adopted a classical Mean Absolute Error (MAE) loss function:

$$\text{MAE} = \frac{1}{W \cdot H \cdot N} \sum_{x,y,n} |\mathcal{Z}(\mathbf{k}_{\mathbf{p}=(x,y)}, \vec{l}_n) - \mathcal{I}_n(x, y)|. \quad (6)$$

## 4 Experimental Results

We started by analysing the behaviour of our proposed MLP model with respect to two relevant parameters, namely the number of PCA bases  $B$  and the parameter  $\sigma$  used to sample the frequencies of the light projection matrix  $\mathbf{B}$ . Then, we quantitatively and qualitatively validated our method with respect to fully-synthetic data as well as with real-world smartphones acquisition.

In all our tests we fixed  $H = 10$  so that the matrix  $\mathbf{B}$  has size  $10 \times 2$  always projecting the input light vector  $\vec{l}$  into a 20-dimensional space. Note that the



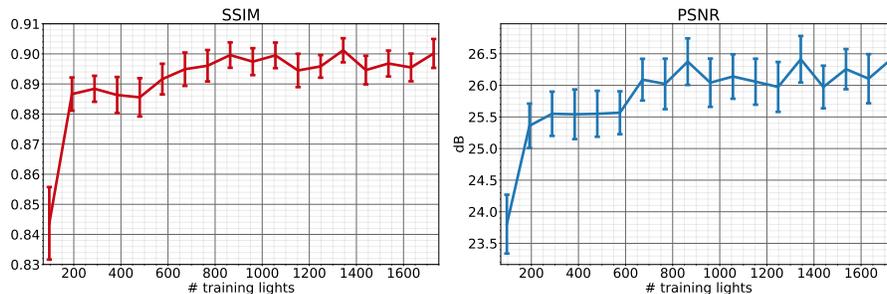
**Fig. 4.** First row: SSIM and PSNR values increasing the number of PCA bases for data compression. Second row: SSIM and PSNR values increasing the sigma.

values of the matrix  $\mathbf{B}$  are randomly sampled before starting the training and never optimised. During the training we used Adam optimiser with a learning rate of  $10^{-3}$  for the first 20 epochs and then reduced to  $10^{-4}$  for another 20.

**Real-world datasets** For the real data we used some coins as test objects, acquired using an Apple iPhone 11 acting as static device and a Samsung Galaxy A40 as the moving one. Videos have been processed as described in Section 3.1 resulting in a MLIC with roughly  $2K$  images for each coin. Then, for each MLIC we randomly selected 25 lights for the test set and discarded the closest light directions (within a radius) so that the learning process is not trained on similar conditions. An example of acquired light directions for the dataset *Coin1* is shown in Figure 3 (left), where the blue dots are the  $\sim 1920$  lights used for training and the red crosses the ones extracted for test.

#### 4.1 Parameter Study

We first studied the effect of the number  $B$  (i.e. PCA bases) on the final relighting quality. Therefore, in the first test we projected the acquired MLIC data into an increasing number of PCA bases, and proceeded with the training process as described. The plots in Figure 4 (first row) show the resulting PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural Similarity Index) for the test set while raising the number of bases  $B$ , from 2 to 32. Note that such results have



**Fig. 5.** Average SSIM and PSNR on test set increasing the number of lights used during training procedure for our method.

been computed by taking the average among all the acquired datasets and repeating the training 10 times due to the random nature of the process. The error bars denote the standard error. We can notice that with  $B = 2$  the relighting quality is quite low, and increasing the number of bases from 3 to 8 corresponds to an increasing reconstruction quality. Both PSNR and SSIM value stabilise for  $B > 8$ , meaning that a higher number of bases would not further improve the output quality. In all our tests we observed that a PCA projection with 8 bases offers a good compression for our smartphone-acquired data. Moreover, we tested the same 8-bases compression for classical RTI datasets where a dome with equispaced lights is used: interestingly, results are numerically and qualitatively better with respect to the autoencoder compression technique as shown in the first row of Table 1.

The next experiment results are shown in the second row of Figure 4: we analysed the relighting quality against the value of  $\sigma$  (on x-axis) used to generate the random values in the matrix  $\mathbf{B}$ . The test was repeated 50 times for each different dataset. We can observe that values around  $\sigma = 0.3$  offer good results in terms of average PSNR and SSIM on the test set, exhibiting also a smaller standard error. As stated in [35],  $\sigma$  is a free parameter that has to be tuned for a particular problem. However, our light directions have unitary norm so, once the optimal  $\sigma$  is defined, it will remain the same regardless the object to reconstruct. Therefore, we used 0.3 in all our real-world tests.

We also tested the effect of the number of acquired light directions (the size of  $N$ ) against the final reconstruction accuracy while keeping the same network layout (Fig.5). This increases the size of the training set but not the storage space required for the model. Both SSIM and PSNR increase with  $N$ , probably because the network can be trained better if a large variety of light conditions can be used. Nevertheless, this increase is almost negligible when the number of light samples exceeds 700 – 800. So, assuming an acquisition in which a carefully planned circular motion around the object is performed, an average video duration of 40 seconds at 25 FPS would be sufficient.

**Table 1.** Relight comparison for different methods.

Dataset	Polynomial		RBF		NeuralRTI		Our	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
SynthRTI	22.7451	0.7932	22.6828	0.8353	26.3658	0.8540	<b>26.4075</b>	<b>0.8553</b>
Coin1	24.0562	0.8643	25.6791	<b>0.9152</b>	25.6846	0.8940	<b>27.0019</b>	0.9118
Coin2	25.8627	0.8798	26.8939	<b>0.9197</b>	26.9147	0.8937	<b>28.0361</b>	0.9105
Coin3	24.2319	0.8642	25.4360	<b>0.9009</b>	25.0304	0.8808	<b>25.7479</b>	0.8975
Coin4	27.6388	0.9155	28.1845	0.9369	27.8950	0.9176	<b>29.6954</b>	<b>0.9398</b>

## 4.2 Comparisons

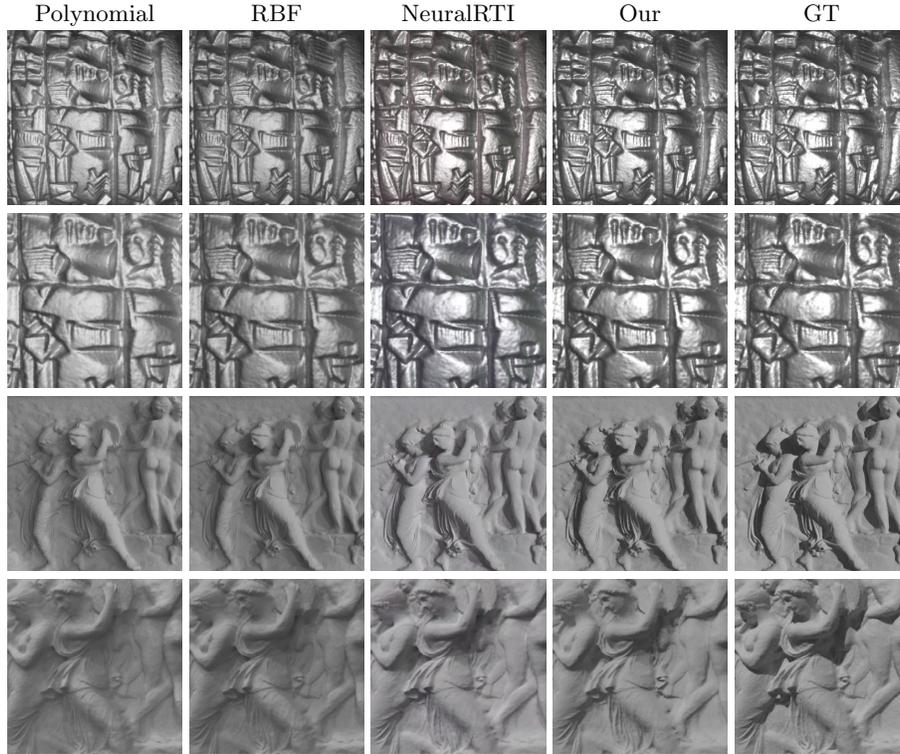
We compared our relighting approach with two classical light interpolation methods, namely polynomial texture maps [14] (from now on, identified as *polynomial*) and RBF. Moreover, we tested against the already discussed learning-based method *NeuralRTI* [5]. We recall that NeuralRTI architecture changes with the number of input lights (the length of the encoder input is  $3N$  because the network works in the RGB space), so we trained it on our acquired data by randomly selecting 100 lights among the training set, since the training process becomes unfeasible for highest input dimensions.

In addition to our data acquired with smartphones, we also validated the method on classical RTI configurations represented by the synthetic dataset presented in [5]. Such data is generated simulating a dome with 69 lights, divided into separate train and test sets of 49 and 20 lights respectively. In all comparisons we set  $B = 8$  (PCA bases),  $\sigma = 0.3$  and  $H = 10$ . Table 1 shows the comparison results. The values in the first row represent the average SSIM and PSNR for the whole SynthRTI dataset. To better evaluate our method comprising not only the reflectance model but also the smartphone-based data acquisition, we show the results for all the objects of the real-world dataset acquired as proposed. Overall, our method exhibits the higher PSNR value, while in some cases relighted data interpolated with RBF give a slightly higher SSIM, but with a significantly smaller PSNR with respect to our method. Note however that RBF is significantly slower in the relighting phase. Also, our values are slightly better with respect to NeuralRTI for the synthetic dataset, where the training lights are sampled uniformly on a dome setup. This indicates that our proposed PCA compression and decoder network still improves the encoder-decoder architecture of [5]. Note that we did not tune any parameter for our results, concluding that the number of PCA bases does not depend on the specific dataset.

Qualitative examples for our acquired dataset are shown in Figure 6, where we display the relighting of three coins with two different test lights (last column shows the ground truth, GT). We can notice that our method is able to recover the object reflectance with high accuracy, especially for the shadows projected near the coins, while the other methods tend to generate light blooms or blurry shadows. Moreover, we notice that NeuralRTI slightly alters the output tint



**Fig. 6.** Relighting comparison of real-world data acquired with two smartphones. The last column (ground truth, GT) shows the actual pictures from the test set.



**Fig. 7.** Qualitative comparison on synthetic data generated with a dome configuration.

with respect to the original: this can be seen in particular in the first two rows. Probably, directly modelling each pixel intensity and colour is more difficult to handle for the network than just the intensity. Using the average UV-value is easier and produces more stable results for non-iridescent objects. Finally, in Figure 7 we show a couple of outputs for the synthetic dataset. Our results are quite similar with respect to NeuralRTI but also in this case our shadow areas are sharper and the images exhibit a higher contrast.

## 5 Conclusions

In this paper we proposed a low-cost technique to perform image relighting on the go using two smartphones for data acquisition. A practical video processing pipeline extracts the MLIC that is compressed and used to train a neural relighting model. Extensive tests in both synthetic and real-world settings show that our network effectively hallucinates images from unseen light directions with high quality. The presented setup can be easily operated directly on the field, with no need of expensive and specialised hardware, allowing researchers to carry out part of their work in an effective and fast way.

## References

1. Ackermann, J., Fuhrmann, S., Goesele, M.: Geometric point light source calibration. In: VMV. pp. 161–168 (2013)
2. Ahmad, J., Sun, J., Smith, L., Smith, M.: An improved photometric stereo through distance estimation and light vector optimization from diffused maxima region. *Pattern Recognition Letters* **50**, 15–22 (2014)
3. Ciortan, I., Pintus, R., Marchioro, G., Daffara, C., Giachetti, A., Gobbetti, E., et al.: A practical reflectance transformation imaging pipeline for surface characterization in cultural heritage (2016)
4. Coules, H., Orrock, P., Seow, C.E.: Reflectance transformation imaging as a tool for engineering failure analysis. *Engineering Failure Analysis* **105**, 1006–1017 (2019)
5. Dulecha, T.G., Fanni, F.A., Ponchio, F., Pellacini, F., Giachetti, A.: Neural reflectance transformation imaging. *The Visual Computer* **36**(10), 2161–2174 (2020)
6. Earl, G., Basford, P., Bischoff, A., Bowman, A., Crowther, C., Dahl, J., Hodgson, M., Isaksen, L., Kotoula, E., Martinez, K., et al.: Reflectance transformation imaging systems for ancient documentary artefacts. *Electronic visualisation and the arts (EVA 2011)* pp. 147–154 (2011)
7. Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F.J., Marín-Jiménez, M.J.: Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition* **47**(6), 2280–2292 (2014)
8. Giachetti, A., Ciortan, I., Daffara, C., Pintus, R., Gobbetti, E., et al.: Multispectral rti analysis of heterogeneous artworks (2017)
9. Giachetti, A., Ciortan, I.M., Daffara, C., Marchioro, G., Pintus, R., Gobbetti, E.: A novel framework for highlight reflectance transformation imaging. *Computer Vision and Image Understanding* **168**, 118–131 (2018)
10. Hartley, R., Zisserman, A.: *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2 edn. (2003)
11. Kinoshita, S., Yoshioka, S., Miyazaki, J.: Physics of structural colors. *Reports on Progress in Physics* **71**(7), 076401 (jun 2008). <https://doi.org/10.1088/0034-4885/71/7/076401>, <https://doi.org/10.1088/0034-4885/71/7/076401>
12. Kinsman, T.: An easy to build reflectance transformation imaging (rti) system. *Journal of Biocommunication* **40**(1) (2016)
13. Kotoula, E., Kyranoudi, M.: Study of ancient greek and roman coins using reflectance transformation imaging. *E-conservation magazine* **25**, 74–88 (2013)
14. Malzbender, T., Gelb, D., Wolters, H.: Polynomial texture maps. In: *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. pp. 519–528 (2001)
15. Manfredi, M., Williamson, G., Kronkright, D., Doehne, E., Jacobs, M., Marengo, E., Bearman, G.: Measuring changes in cultural heritage objects with reflectance transformation imaging. In: *2013 Digital Heritage International Congress (Digital-Heritage)*. vol. 1, pp. 189–192. IEEE (2013)
16. Manrique Tamayo, S.N., Valcárcel Andrés, J.C., Osca Pons, M.: Applications of reflectance transformation imaging for documentation and surface analysis in conservation. *International Journal of Conservation Science* **4**, 535–548 (2013)
17. Mudge, M., Malzbender, T., Chalmers, A., Scopigno, R., Davis, J., Wang, O., Gunawardane, P., Ashley, M., Doerr, M., Proenca, A., et al.: Image-based empirical information acquisition, scientific reliability, and long-term digital preservation for the natural sciences and cultural heritage. *Eurographics (Tutorials)* **2**(4) (2008)

18. Mudge, M., Malzbender, T., Schroer, C., Lum, M.: New reflection transformation imaging methods for rock art and multiple-viewpoint display. In: Ioannides, M.; Arnold, D.; Niccolucci, F. & Mania, K., eds., *The 7th International Symposium on Virtual Reality, Archaeology and Cultural Heritage*. vol. 6, pp. 195–202. Vast (2006)
19. Mytum, H., Peterson, J.: The application of reflectance transformation imaging (rti) in historical archaeology. *Historical Archaeology* **52**(2), 489–503 (2018)
20. Otsu, N.: A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics* **9**(1), 62–66 (1979). <https://doi.org/10.1109/TSMC.1979.4310076>
21. Palma, G., Corsini, M., Cignoni, P., Scopigno, R., Mudge, M.: Dynamic shading enhancement for reflectance transformation imaging. *Journal on Computing and Cultural Heritage (JOCCH)* **3**(2), 1–20 (2010)
22. Pintus, R., Dulecha, T.G., Ciortan, I., Gobbetti, E., Giachetti, A.: State-of-the-art in multi-light image collections for surface visualization and analysis. In: *Computer Graphics Forum*. vol. 38, pp. 909–934. Wiley Online Library (2019)
23. Pistellato, M., Albarelli, A., Bergamasco, F., Torsello, A.: Robust joint selection of camera orientations and feature projections over multiple views. vol. 0, p. 3703 – 3708 (2016). <https://doi.org/10.1109/ICPR.2016.7900210>
24. Pistellato, M., Bergamasco, F., Albarelli, A., Torsello, A.: Dynamic optimal path selection for 3d triangulation with multiple cameras. vol. 9279, p. 468 – 479 (2015)
25. Pistellato, M., Bergamasco, F., Albarelli, A., Torsello, A.: Robust cylinder estimation in point clouds from pairwise axes similarities. p. 640 – 647 (2019). <https://doi.org/10.5220/0007401706400647>
26. Pitard, G., Le Goïc, G., Mansouri, A., Favrelière, H., Desage, S.F., Samper, S., Pillet, M.: Discrete modal decomposition: a new approach for the reflectance modeling and rendering of real surfaces. *Machine Vision and Applications* **28**(5), 607–621 (2017)
27. Ponchio, F., Corsini, M., Scopigno, R.: Relight: A compact and accurate rti representation for the web. *Graphical Models* **105**, 101040 (2019)
28. Porter, S.T., Huber, N., Hoyer, C., Floss, H.: Portable and low-cost solutions to the imaging of paleolithic art objects: A comparison of photogrammetry and reflectance transformation imaging. *Journal of Archaeological Science: Reports* **10**, 859–863 (2016)
29. Rainer, G., Jakob, W., Ghosh, A., Weyrich, T.: Neural btf compression and interpolation. In: *Computer Graphics Forum*. vol. 38, pp. 235–244. Wiley Online Library (2019)
30. Ren, P., Dong, Y., Lin, S., Tong, X., Guo, B.: Image based relighting using neural networks. *ACM Transactions on Graphics (ToG)* **34**(4), 1–12 (2015)
31. Schuster, C., Zhang, B., Vaish, R., Gomes, P., Thomas, J., Davis, J.: Rti compression for mobile devices. In: *Proceedings of the 6th International Conference on Information Technology and Multimedia*. pp. 368–373. IEEE (2014)
32. Shen, C.: Analysis of detrended time-lagged cross-correlation between two nonstationary time series. *Physics Letters A* **379**(7), 680–687 (2015). <https://doi.org/https://doi.org/10.1016/j.physleta.2014.12.036>, <https://www.sciencedirect.com/science/article/pii/S0375960114012766>
33. Smys, S., Chen, J.I.Z., Shakya, S.: Survey on neural network architectures with deep learning. *Journal of Soft Computing Paradigm (JSCP)* **2**(03), 186–194 (2020)
34. Suzuki, S., be, K.: Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing* **30**(1),

- 32–46 (1985). [https://doi.org/https://doi.org/10.1016/0734-189X\(85\)90016-7](https://doi.org/https://doi.org/10.1016/0734-189X(85)90016-7), <https://www.sciencedirect.com/science/article/pii/0734189X85900167>
35. Tancik, M., Srinivasan, P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J., Ng, R.: Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems* **33**, 7537–7547 (2020)
  36. Uribe, M.D.G., Wheatley, D.W.: Rock art an digital technologies: the application of reflectance transformation imaging (rti) and 3d laser scanning to the study of late bronze age iberian stelae. *Menga: Revista de prehistoria de Andalucía* (4), 187–203 (2013)
  37. Vieira, M., Guimarães, P.V., Violante-Carvalho, N., Benetazzo, A., Bergamasco, F., Pereira, H.: A low-cost stereo video system for measuring directional wind waves. *Journal of Marine Science and Engineering* **8**(11), 831 (2020)
  38. Watteuw, L., Hameeuw, H., Vandermeulen, B., Van der Perre, A., Boschloos, V., Delvaux, L., Proesmans, M., Van Bos, M., Van Gool, L.: Light, shadows and surface characteristics: the multispectral portable light dome. *Applied Physics A* **122**(11), 1–7 (2016)
  39. Xu, Z., Sunkavalli, K., Hadap, S., Ramamoorthi, R.: Deep image-based relighting from optimal sparse samples. *ACM Transactions on Graphics (ToG)* **37**(4), 1–13 (2018)